Towards Realizing Melt-electrospun Bone Scaffolds: Theoretical Modeling and Simulation of Melt-Electrospinning as a Multi-body System

by

Abiy Tadesse Wubneh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering
University of Alberta

Abstract

There has been a growing interest in integrating melt-electrospinning (MES) with three-dimensional (3D) fabrication methods to produce constructs with controlled internal micro-architectures. One application area that could benefit from this integration is bone tissue engineering (TE). Due to the close functional and micro-architectural similarities between the melt-electrospun fiber depositions and the extracellular matrix (ECM) of the natural bone, melt-electrospun constructs have the potential to serve as ideal artificial environments for initial cell attachment and differentiation. The internal micro-structure of the construct, including the spatial distribution of accumulation densities and average fiber diameters, could be customized by controlling the processing parameters.

There are, however, significant technological gaps hindering the realization of this approach. Among them is the need for reliable methods for predicting the behavior of the process under a set of material and processing conditions. Available predictive models lack the necessary features to describe the process in its entirety, and their scope is usually limited to describing only a subset of the process. For example, while some models describe the jet initiation mechanism and Tailor's cone formations, others focus only on characterizing the stable jet region. There are also existing models that describe the unstable region but are either based on oversimplified assumptions or do not consider some critical dynamical behaviors of the system affecting the fiber morphology.

A theoretical predictive model is proposed in the present study to address these shortcomings. The proposed model aims to simulate the MES process under different material and processing conditions, including its potential use in simulating direct-writing and 3D printing scenarios. By treating the fiber as a serially connected multi-body system, the fiber dynamics are formulated using Kane's and Udwadia-Kalaba's methods to take advantage of their suitability for multi-body systems with large numbers of components and complex constraints. The model formulation includes several advanced features, including Maxwell's generalized standard linear solid (SLS) for improved viscoelastic behavior representation. The anchoring effects of the collector plate and independent motions of the spinneret and the collector plate are also included in the problem formulation to enable the simulation of direct-writing and 3D printing scenarios.

An experimental study is carried out to validate the model. The simulated results and the experimental observations are compared to assess the model's predictive accuracy. The comparison revealed good agreement between the two, indicating the model's reliability. In addition, as a byproduct of the investigation, two regression models are extracted from the observation data for predicting the collection and average fiber diameters, respectively. However, these regression models are limited to only stationary and vertical melt-spinning scenarios using polylactic acid (PLA) material.

A separate experimental investigation has been conducted to demonstrate the model's capability in predicting direct-writing scenarios in which the melt-electrospun fibers are deposited based on predefined patterns. To test this new capability, the predictive model and an MES printer were run using the same toolpath (G-code) data and processing conditions. The results show reliable predictive accuracy, indicating that the model could be used for analyzing fiber behaviors.

Overall, this study presents a new theoretical predictive model for simulating the MES process, which has the potential to overcome the limitations of currently available tools. The model predicts with reasonable accuracy the deposition characteristics of melt-electrospun fibers under different material and processing conditions. It can also be used to simulate direct-writing scenarios. With further marginal improvement, the predictive model could accelerate the application of MES in 3D printing applications.

Preface

This thesis is an original work by Abiy Wubneh, conducted under the supervision of Dr. Cagri Ayranci and Dr. Chun-Il Kim at the Department of Mechanical Engineering, University of Alberta.

The literature review material presented in Chapter 2 is part of a peer-reviewed journal published as *Wubneh A, Tsekoura EK, Ayranci C, Uludag H. "Current state of fabrication technologies and materials for bone tissue engineering." ACTA BIOMATERIALIA. 2018;80:1-30.* Most of the contribution made by Tsekoura EK. and Dr. Uludag H., which is mainly biology-specific subject matter, has been redacted from the thesis not to distract from the main topic of the thesis, which focuses on the theoretical modeling of MES. Most of the remaining material in the chapter, including all figures except for Fig. 13 and Fig. 14, was originally written and created by Abiy Wubneh and later edited and improved upon by the co-authors.

A version of Chapter 4 is published as *Wubneh A, Ayranci C, Kim C-I. "A Novel Theoretical Model Development and Simulation of Melt-Electrospinning Using Kane's and Udwadia-Kalaba Methods." ADVANCED THEORY AND SIMULATIONS. January 2022:2100278.* Abiy Wubneh was primarily responsible for developing the mathematical formulation and writing the first manuscript draft under the guidance of the co-authors and supervisors, Dr. Cagri Ayranci and Dr. Chun-Il Kim. The co-authors/supervisors have been involved with concept formation and manuscript composition.

The material presented in Chapter 5 will be submitted in its entirety for publication under the title: *A Study on Theoretical Predictive Model and Experimental Findings of Melt-electrospinning Process*. The work, including the modification of the commercial 3D printer into a MES apparatus, was done by Abiy Wubneh with the guidance of supervisors Dr. Cagri Ayranci and Dr. Chun-Il Kim. The supervisors were also involved with experimental planning and manuscript composition.

Abiy Wubneh, Dr. Cagri Ayranci, and Dr. Chun-Il Kim plan to publish the material in Chapter 6 shortly after graduation.

Abiy Wubneh wrote all the MATLAB®codes needed for the simulation of MES. These codes are listed in Appendix A.

To my Father, Tadesse Wubneh

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Cagri Ayranci, for his support, guidance, and patience throughout my Ph.D. journey. His belief in my research and valuable insights have been instrumental in the successful completion of my thesis.

I would also like to extend my sincere thanks to my co-supervisor, Dr. Chun-Il Kim, for his continuous support, encouragement, and valuable feedback throughout my research. His contributions have significantly shaped my work and helped me achieve my research goals.

My sincere appreciation also goes to Dr. Hassan Uludag, a supervisory committee member, for his feedback and valuable suggestions that have improved my research work, especially my first paper.

I am also grateful to my colleague and fellow Ph.D. candidate, Jiawei Chen, for her support, collaboration, and laboratory training. Her patience and willingness to answer my endless questions have been immensely helpful.

I would also like to thank my friend and ex-group member, Dan Aldrich, for creating this beautiful LaTeX thesis template, which made my work easy. His effort has benefitted many UofA graduate students, and we are forever grateful to him.

My family has been my pillar of strength throughout this journey. Words cannot express my gratitude to my mother, Emodi, for her prayers and unwavering

support. To Mele, Astu, Fidi, TG, Kuku, and Brad, your support has always been invaluable, and I cannot thank you enough for being there for me.

I express my heartfelt appreciation to my partner, Joanna, for her unwavering support, encouragement, and understanding throughout my Ph.D. journey. Her support and willingness to listen to my ideas and thoughts have been a source of strength for me.

Lastly, I would be remiss in not mentioning my dog, *His Majesty, First of His Name, Conqueror of the Living Room*, King Gouda. He has been a source of comfort and instrumental in lowering my stress levels during this journey.

Table of Contents

List of Tables

List of Figures

List of Symbols

Greek Symbols

ε	Strain
$\boldsymbol{\Gamma}$	Torque
$\boldsymbol{\omega}$	Angular velocity
$\boldsymbol{\theta}, \theta$	Angle
$\boldsymbol{\varphi}$	Vector of constraint equations
η	Viscosity
η^*	Complex viscosity
γ	Surface tension
κ_τ	Torsional spring stiffness
σ	Stress
σ_Y	Tensile strength (yield)
φ	Constraint equation

Latin Symbols

m	Mass
$\hat{\boldsymbol{e}}$	Euler parameters
$\boldsymbol{\mathcal{F}}$	Force

$\boldsymbol{A}$	Coefficient matrix, constraint equations	
$\boldsymbol{a}$	Acceleration	
$\boldsymbol{B}$	Damping coefficient matrix	
$\boldsymbol{b}$	Constraint equations vector	
$\boldsymbol{E}$	Electric field	
$\boldsymbol{E}$	Quaternion rotation matrix	
$\boldsymbol{F}$	Generalized fores	
$\boldsymbol{f}$	Force	
$\boldsymbol{G}$	Quaternion rotation matrix	
$\boldsymbol{g}$	Acceleration due to gravity	$9.81\,\mathrm{m\,s^{-1}}$
$\boldsymbol{H}$	Angular momentum	
$\boldsymbol{I}$	Identity matrix	
$\boldsymbol{J}$	Moments of inertia tensor	
$\boldsymbol{K}$	Stiffness matrix	
$\boldsymbol{M}$	Mass matrix	
$\boldsymbol{Q}$	Impressed forces	
$\boldsymbol{q}$	Unit quaternion	
$\boldsymbol{Q}^c$	Virtual constraint forces	
$\boldsymbol{R}$	Force	
$\boldsymbol{r}$	Position vector	
$\boldsymbol{T}$	Generalized moments	
$\boldsymbol{u}$	Vector of generalized coordinates	
$\boldsymbol{v}$	Imaginary components of a quaternion	
$\boldsymbol{v}$	Velocity	

$\dot{Q}$	Flow rate	
$\hat{\mathbf{a}}$	Coordinate system basis	
$\hat{\mathbf{b}}$	Coordinate system basis	
$\hat{\mathbf{n}}$	Coordinate system basis	
$\mathcal{B}_i$	Rigid body label	
$c_{\tau i}$	Torsional damping coefficient	
D	Diameter	
d	Diameter	
E	Elastic Constant (Young's Modulus)	
F	Generalized forces (scalar)	
G''	Loss modulus	
G'	Storage modulus	
h	Collector distance	
J	Polar moment of inertia	
k_e	Coulomb constant	$8.987 \times 10^9\,\mathrm{Nm}^2/\mathrm{C}$
q	Electric charge	
R	Rotation matrix	
T	Generalized moments (scalar)	
T	Temperature	
V	Voltage	
w	Real component of a quaternion	
t	Time variable	

Math Symbols

$\mathbb{C}$	Complex number space	
$\mathbb{H}$	Hamiltonian space	
$\mathbb{R}$	Real number space	
μ	Mean	
π	Constant	3.14159
σ	Standard deviation	
R	Coefficient of determination	

Abbreviations

2D two-dimensional
3D three-dimensional

AM additive manufacturing
AMF additive manufacturing file
ANOVA analysis of variance

CAD computer-aided design
CAM computer-aided manufacturing
CAWS computer-aided wet spinning
CDLP continuous digital light processing
CNC computer numerical control
CSBD critical-sized bone defects
CT computed tomography

DAE differential-algebraic equations
DLP digital light processing
DMLS direct metal laser melting
DOE design of experiments

EBM electron beam melting
ECM extracellular matrix

FDM fused deposition modeling
FEA finite elements method
FFF fused filament fabrication

GF gas foaming

HA hydroxyapatite

MES melt-electrospinning

MFCL Multi-functional Composite Laboratory
MRI magnetic resonance imaging
MSC mesenchymal stem cells

NFES near-field electrospinning

OGP steogenic growth peptide

PA polyamide
PBF powder bed fusion
PCL Polycaprolactone
PDLLA Poly-D-L-lactic acid
PEEK polyether-ether-ketone
PEG polyethylene glycol
PLA polylactic acid
PLGA Polylactide-co-glycolic acid
PLLA Poly-L-lactic acid
PPF Polypropylenefumarate
PS polysaccharides
PVA polyvinyl alcohol

ROM rule of mixture

SES solution-electrospinning
SFF solid free-form fabrication
SLM selective laser melting
SLS standard linear solid
SLS selective laser sintering
STL stereolithography

TCP tricalcium phosphate
TE tissue engineering

UK Udwadia-Kalaba
UofA University of Alberta
UV ultraviolet

VMR variance-to-mean ratio

Chapter 1

Introduction

There has been a growing demand for technologies to advance the treatment of critical-sized bone defects (CSBD) caused by injuries or other medical complications [1]. This quest, compounded by the advent of functionally-tailored, biocompatible, and biodegradable materials, has ignited an enormous research interest in bone tissue engineering, a practice in which new functional bone tissues are regenerated from existing bone cells. The regeneration process typically occurs inside artificially created environments known as *bone scaffolds*.

Some notable challenges and gaps in existing fabrication technologies must be resolved before realizing the full potential of bone tissue engineering. Specifically, the existing fabrication methods can only produce bone scaffolds that meet some clinical requirements, limiting their use only to a few practical applications.

The melt-electrospinning (MES) process is a potential fabrication method that could address some of these challenges. The inter-fiber architectures of depositions produced by this process closely resemble the extracellular matrix (ECM) of the natural bone tissue [2]. It is also possible to impregnate the material used in the process with drugs and growth factors that could be released at the implant site to promote the health and growth of the newly regenerated bone tissue.

Despite these advantages, this method faces its challenges. The correlations between its process parameters and the resulting fiber characteristics are yet to be thor-

oughly understood. A processing guideline is needed for selecting optimized parameter settings to produce fiber accumulations of targeted topological and morphological characteristics. This capability is vital to deploying MES in bone tissue engineering applications with stringent geometrical and morphological requirements.

This research addresses some of these gaps by proposing a theoretical predictive model. The goal is to deploy this model to simulate the MES process under a given set of material and processing conditions and predict the characteristics of the resulting melt-electrospun fiber and finally, produce the intended structure.

1.1 Background

Tissue engineering (TE) relies on the use of *tissue scaffolds* for housing and facilitating the growth of the target tissues [3–6]. In the context of bone tissue engineering, the scaffolds are generally three-dimensional (3D) mechanical structures with predefined internal micro-architectures and chemical compositions. The primary objective of a bone scaffold is to provide structural support while maintaining a conducive environment for guided regeneration and growth of bone tissue at an implant site. The ideal scaffold must have good mechanical strength, osteoconductivity, biocompatibility, and biodegradability properties. It is also characterized by optimally-sized and interconnected porous spaces to provide room for new cell formation, and routes for delivery of nutrients and removal of metabolic wastes from the implant site [7].

Many traditional and free-form fabrication methods, such as gas foaming, cryogelation, and 3D printing, have been investigated for bone scaffold applications and are occasionally commercialized for clinical use [2, 8, 9]. While most of these methods could not produce internal micro-structures meeting the clinical requirements, others needed mechanisms to control the spatial distributions of the micro-structures inside the scaffold [10]. The efforts, regardless, have led to promising results and a deeper understanding of the geometrical, mechanical, chemical, and biological requirements associated with bone scaffolds.

One of the fabrication methods considered is MES. Electrospinning, in general, is a fabrication process in which a steady stream of electrically charged viscous polymer is

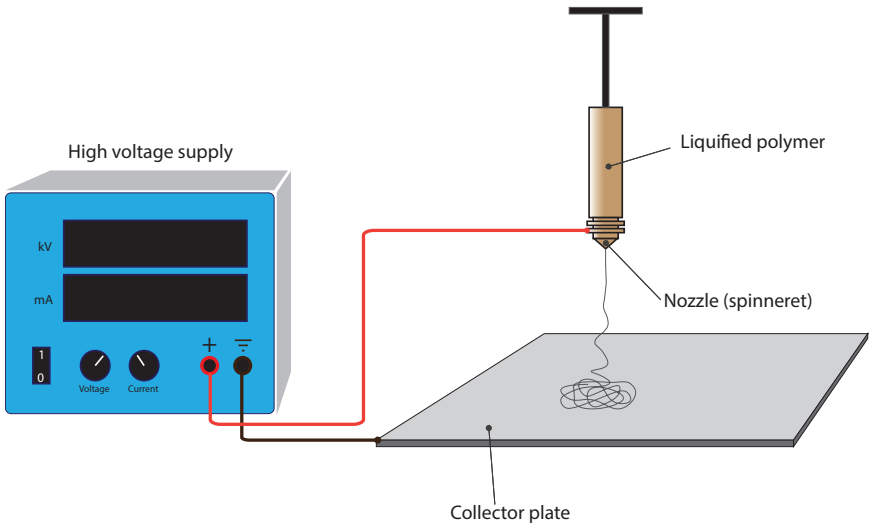


Figure 1.1: Basic electrospinning setup

drawn into a fiber under the actions of electrostatic forces [11]. As shown in Fig. 1.1, the nozzle and the collector plate are typically connected to the opposite terminals of a high-voltage, low-current power supply unit. Due to the ensuing electrostatic charge and electric field, Coloumb and electric field forces are exerted on the charged liquid polymer, pulling and accelerating material from the nozzle tip toward the collector plate. This action draws the polymer into a continuous fiber with diameters ranging from 5 to 500 µm, depending on the mechanism employed to liquefy the polymer. If chemical solvents are employed to dissolve the polymer into the viscous state, the process is referred to as *solution-electrospinning* (SES). Whereas if a thermal energy source is used instead to change the state of the polymer, the resulting process is known as melt-electrospinning (MES) [12–14]. The absence of toxic chemicals in MES makes the process an ideal fabrication method for articles used in clinical applications such as wound dressing [15–20].

Integrating the solid free-form fabrication (SFF) technology and the MES brings a unique capability to bone tissue engineering [21]. As depicted in Fig. 1.2, SFF permits the deposition of the melt-electrospun fiber into intricately shaped constructs at the macro level. Meanwhile, because of the inherent fiber characteristics, the MES furnishes the inside of the scaffold with ECM-like micro-architectures. An example scenario where the effectiveness of this technique could be demonstrated is where the

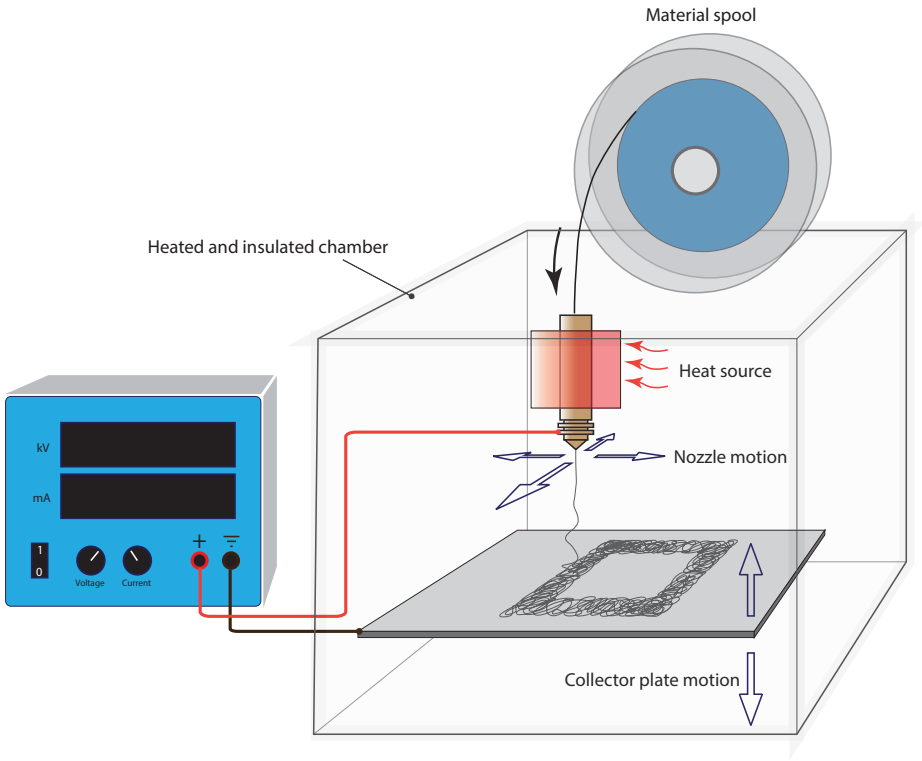


Figure 1.2: Multi-axis controlled MES writing

geometrical data of the scaffold is generated by directly scanning and digitizing the natural bone tissue from the CSBD-affected area of the patient. The digitized data could later be used to 3D print a bone scaffold to replace the affected bone tissue.

While this approach provides a powerful mechanism to match the general shape of the scaffold to the shape of the original bone tissue, challenges remain regarding how to define and control the spatial distribution of the micro-architectures inside the printed scaffold. This problem must be addressed because the type and degree of vascularization, which in turn dictate the kind of bone tissue formation, depend on the internal geometrical architecture of the scaffold [22, 23]. For example, regenerating fibrous tissues requires smaller pore sizes ($<$75 µm). In contrast, unmineralized and fully mineralized bone tissue formations require intermediate (75–100 µm) and larger (200 µm) pore sizes, respectively [24, 25]. Vascularization is also affected by scaffold roughness, internal connectivity, porosity levels, chemical composition, and material biocompatibility.

1.2 Research Problem

Understanding the interrelationships among MES's input and output process parameters, as well as the effects of the operating conditions, is crucial to creating a construct with predefined internal micro-architectures [26, 27]. However, precise prediction of the fiber characteristics could only be realized using advanced predictive mathematical models. While such forward predictive capability is desirable and a step in the right direction, it alone is inadequate to produce scaffolds with predefined internal characteristics. A reverse prediction method is ultimately needed to estimate the optimal processing parameter values for a desired fiber characteristics outcome. Such a predictive model will be a convenient tool for directly mapping clinical requirements to a set of compatible processing parameter values.

Different groups have made analytical modeling efforts toward this goal. The models reviewed focus on conceptualizing the fiber as an overhanging viscoelastic pendulum. The total length of the fiber is usually discretized into a finite number of smaller segments. Viscoelasticity is imparted to the models using rheological models as connective elements between successive segments. The time histories of the fibers are predicted from the numerical integration of the equations of motion of the overall systems, mainly under the actions of gravitational, viscoelastic, surface tension, and Coulomb forces. Although these models have laid a strong foundation for future developments, they are too simplistic to be reliable predictive tools for a few reasons. First, most of them are specifically developed with SES in mind, and the effects of temperature, a hallmark of MES, are generally not considered. Second, they are based on oversimplified assumptions about the viscoelastic properties of the materials. It typically takes a 10–15 layer generalized Maxwell viscoelastic rheological model to curve-fit experimental data with reasonable accuracy. However, most of the models reviewed employ only a single layer of Maxwell link [28]. This is suspected of introducing errors that lead to a sub-optimal representation of creep and relaxation behaviors of materials [29]. Another implicit but consequential assumption in the models' formulation is that one end of the fiber would always remain freely wiggling in the air without touching the collector plate. As a result, the anchoring effects of the

collector plate on the movement of the fiber are ignored entirely. This is problematic, especially in the simulation of near-field electrospinning (NFES) scenarios, wherein the distance between the nozzle and the collector plate is very short, and the fiber often buckles under the support of the collector plate [30].

Additionally, if the spinneret and the collector plate were to have their distinct motions (as is commonly the case in most 3D printing scenarios), in addition to the anchoring effects of the collector plate (print bed), external momentum would be imparted to the fiber by the moving platforms. This phenomenon alters the dynamic behavior of the fiber and, by extension, its deposition characteristics.

1.3 Research Aim

This research proposes a theoretical model for predicting the fiber characteristics of the MES process for a given set of materials and processing parameters. Specifically, the research objectives are to:

1. develop and implement a theoretical forward-predictive model to simulate the MES process in both the transient and steady-state scenarios
2. improve the viscoelastic property representations of the materials in the model
3. include the effects of the fiber sticking on the collector plate/drum in the theoretical formulation
4. include the effects of the nozzle and collector plate motions in the model, and
5. develop an algorithm for continuously simulating the MES process without the need to predefine the model size

1.4 Significance

Although the proposed model builds upon the contributions of other existing models, the commonality between the proposed and the previous models stops at conceptualizing the fiber as a discretized multi-body system. Several improvements and advanced new features are introduced in the proposed model to address the shortcomings mentioned above. Further development and implementation of the multi-body system approach, including the addition of the platforms' movements and the collec-

tor plate's anchoring effects into the problem formulation, are novel contributions of this research. Specifically, the novelties are:

1. the inclusion into the dynamic formulation of the anchoring and whipping dampening forces generated due to the fiber sticking on the collector plate
2. the inclusions of the motions of the spinneret and the collector plate in the formulation to allow simulation of direct-writing and 3D printing scenarios
3. the use of Maxwell's standard linear solid (SLS) rheological links as viscoelastic joints for improved representation of creep and relaxation behaviors of the molten polymer
4. addition of rotational viscoelastic elements into the model to capture the whipping motion-dampening phenomenon
5. consideration of rotational degrees of freedoms of the discretized fiber segments in the formulation, and
6. the introduction and computer implementation of a computationally efficient algorithm for continuous simulation of the MES processes without predefining the model size or allocating memory ahead of time.

The formulation of the dynamic equations incorporates most of the previously overlooked dynamic fiber behaviors into the model. Like the early models, the fiber is mathematically treated as a serially linked multi-body system composed of shorter fiber segments. First, the unconstrained equations of motion of the multi-body system are formulated using Kane's Dynamical Equations (also known as *Kane's Method*). This method, which has roots in the Lagrangian mechanics but does not require Lagrange multipliers, is particularly beneficial for large-scale multi-body systems, such as the problem at hand. The absence of the Lagrange multiplier makes Kane's Method an efficient and computer-friendly option for dealing with dynamical systems with a larger number of generalized coordinates (or degrees of freedom). It substantially simplifies the modeling process and reduces the computational load during simulations.

There are many constraints in the multi-body system arising from the viscoelasticity of the material, the movements of the platforms, the fiber-platform sticking

phenomenon, and the use of quaternions for orientation representation. Constraints in the system are handled using the Udwadia-Kalaba (UK) Method, which allows the conversion of each constraint equation into an equivalent constraint force. The constraint forces are added to the unconstrained equations of motion as regular force vectors to furnish the constrained equations of motion of the overall system.

The combined use of the above two methods in the analytical model formulation is a notable departure from previous endeavors that heavily relied on the classical Lagrange's Method. Because of the benefits derived from this approach, the proposed model is expected to simulate the complete process from the jet initiation at the nozzle's tip to the fiber's deposition on the collector plate. Furthermore, the model is designed to simulate 3D printing and direct-writing scenarios by defining the nozzle and collector plate paths as kinematic constraints.

The proposed model generates a large amount of data during simulation that could be post-processed further to gain meaningful insight into the fiber deposition characteristics. The raw simulation result primarily contains the time history of each fiber segment in the multi-body system, including the position, orientation, linear and angular velocities, accelerations, forces, and moments for every bead. As will be demonstrated in Chapter 5, this data could be further post-processed to extract important topological and morphological parameters of the spun fibers. The extracted information may include average fiber diameter, collection diameter, collection pattern, collection density, and other parameters that may be of practical interest when adapting MES as a bone scaffold fabrication method. These parameters dictate the internal micro-structure of 3D printed scaffold constructs. The ability to predict their spatial distribution means predicting the type of bone tissue that could be generated using scaffolds made from the process.

1.5 Structure of the Thesis

Chapter 1 introduced the concept of tissue engineering. This was followed by a brief discussion highlighting the benefits and challenges of employing MES as a potential bone scaffold fabrication method. The research novelties and objectives were also

outlined. The chapter concluded by highlighting some challenges and limitations of the research.

Chapter 2 is a literature review of the current state of materials and fabrication technologies used in bone tissue engineering. A review of the existing analytical and experimental models is also presented to provide context for the current work.

Chapter 3 overviews the fundamental theories and principles applied in the upcoming chapters, where the analytical formulation and experimental characterization are discussed. Topics examined here cover the theories of viscoelasticity, kinematics, multi-body dynamics, and constraints. While no novel material is introduced here, this chapter intends to serve as a one-stop reference source for fundamental theories and methods used in the analytical model formulation.

Chapter 4 is entirely dedicated to the analytical formulation of MES as a multi-body system—the novelty of the research. The equations of motions of the fiber are analytically developed here by treating the fiber as a serially connected multi-body system. Relevant constraints representing different events in the process are identified and incorporated into the equations. A computer implementation algorithm is proposed to enable continuous simulation without the need to predefine the model size or allocate computer memory.

Chapter 5 focuses on the experimental aspects of the research. Two types of experiments are discussed here. The first one concerns the material characterization of Polylactic acid (PLA)—the material of choice for testing the model. A rheological test is first carried out on this material to extract its elastic parameter values. The second experiment is a comparative study designed to validate the theoretical model. While the values of the elastic parameters extracted from the first experiment are used in the theoretical model simulation, an actual MES is carried out using the same material. The observed fiber characteristics are compared with the simulation results to evaluate the predictive performance of the proposed model.

Chapter 6 demonstrates the application of the theoretical model in simulating direct-writing applications. The effects of translational nozzle speeds, commonly re-

ferred to as feed rates, on the fiber characteristics are investigated using the simulation tool as well as with experiments.

Finally, a closing discussion on the major results and highlights of the study is provided in Chapter 7. The current model's limitations are also discussed in this chapter to identify areas for future improvement.

Chapter 2

Literature Review: Current State of Fabrication Technologies and Materials for Bone Tissue Engineering

Chapter Abstract

A range of traditional and solid free-form fabrication (SFF) technologies have been investigated and, on numerous occasions, commercialized for use in regenerative tissue engineering (TE). The demand for technologies capable of treating critical-size bone defects has been on the rise. This quest, accompanied by the advent of functionally tailored, biocompatible, and biodegradable materials, has generated enormous research interest in bone TE. As a result, different materials and fabrication methods have been investigated towards this end, leading to a deeper understanding of the geometrical, mechanical, and biological requirements associated with bone scaffolds. As our understanding of scaffold requirements expanded, so did the capability requirements for fabrication processes. This review aims to broadly examine existing scaffold fabrication processes and highlight future trends in their development. First, a brief review of the natural biological process by which bone tissue regenerates itself is presented to appreciate the clinical requirements for bone scaffolds. This

discussion is followed by a summary and comparison of commonly used implant techniques to highlight the advantages of TE-based approaches over traditional grafting methods. A detailed discussion of the clinical and mechanical requirements for bone scaffolds follows. The remainder of the manuscript is dedicated to current scaffold fabrication methods, their unique capabilities, and perceived shortcomings. The range of biomaterials employed in each fabrication method is summarized. Selected traditional and non-traditional fabrication methods are discussed, highlighting their future potential from the authors' perspective. This study is motivated by the rapidly growing demand for effective scaffold fabrication processes to economically produce constructs characterized by complex internal and external architectures.

2.1 Background on Bone Tissue and Repair

Understanding the physiology of the natural bone is the first step in developing a successful scaffold for bone repair. Bone, or the osseous tissue, is a living structure that serves multiple vital bodily functions. It is mainly made up of two types of structural tissues, namely cancellous (trabecular) and cortical bone [24, 31] (Fig. 2.1). The cancellous tissue constitutes the inner, softer, lighter-density, and highly vascularized core of the bone mass (50–90 vol%), while the cortical tissue is mainly the outermost and denser boundary (10 vol%). The cancellous bone, with compressive strength in the range of 7–10 MPa, is characterized by a lower elastic modulus [25]. This highly porous (30–90%) structure is where most metabolic activities occur [32]. The porosity of the cancellous tissue depends on local cellular activity, and the mediators that affect cellular activity may alter the trabecular structures that influence pore size. Cancellous tissue houses blood vessels, bone marrows, and many other biological ingredients and allows the transportation of nutrients and metabolic waste. Because of the presence of organic content in its structure, this tissue is characterized by superior elasticity and higher levels of resilience and toughness. However, the tissue is also characterized by an insufficient tensile strength ($\sigma_Y < 2\,\text{MPa}$) [33]. Due to its higher mineralization content and absence of organic matter, the cortical tissue possesses

higher elastic modulus, high stiffness, and low toughness [34]. The harder and denser structure of cortical tissue (5–30% porosity) serves as a protective enclosure for the inner fragile cancellous tissue. With a compressive strength range of 130–225 MPa (σ_Y: 60–160 MPa, E: 3–30 GPa) [35], cortical bone plays the role of the main structural support element in the body. The combination of these two tissues makes bone a unique mechanical structure with the capacity to withstand higher levels of loads and deformations than could have been achieved individually by its constituents [32]. About 70% of bone comprises hydroxyapatite (HA) crystals, a member of the calcium phosphate mineral family [32]. The remaining 20–30% is a mixture of water, members of the collagen family of proteins, and other proteins and proteoglycans that make up the organic portion of the bone mass [36, 37]. Bone-resident cells represent only about 2% of this organic matrix [34].

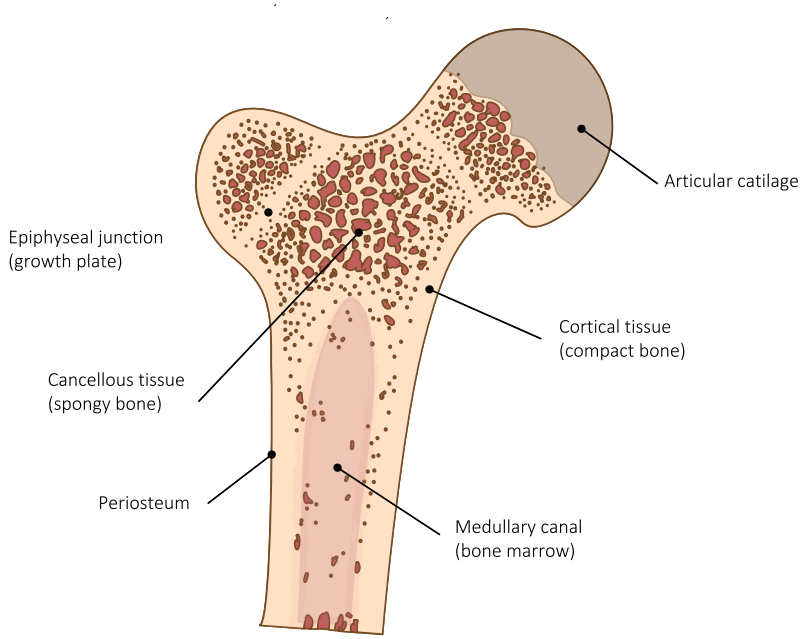


Figure 2.1: Schematic diagram of typical bone tissue

2.2 Interventions for Bone Repair

In the event of excessive damage to the bone tissue due to an accident or degenerative disease, the natural bone (*remodeling*) process alone might not be sufficient to restore

bone integrity. These types of damages may generally require an intervention in the form of *autografting* or *allografting* [38, 39]. In autografting, the affected area is supplemented with bone tissue extracted from the patient's body. This method is well known for its relatively higher success rate because of guaranteed biocompatibility and low risk of rejection [40, 41]. The limitations of this technique stem from its suitability for treating only relatively small defects and the need for additional surgical procedures at donor sites. Critical-size bone defects (CSBD) require more donor tissue that might not be readily available from the patient's body. This challenge limits the applicability of this technique to only the treatment of small-size defects [41, 42]. *Allografting*, which is the process of grafting using processed bone tissue collected from other donors, has been employed in situations wherein the grafting material cannot be harvested from the patient for various reasons. Compared to autografting, this method poses some additional challenges and risks. Preserving the osteopotency of the material during processing is the primary challenge reported. There is also the risk of donor-to-recipient infection transmission and implant rejection.

Tissue engineering (TE) is an alternative approach to generating bone grafts. It involves using temporary artificial environments specifically designed to facilitate the regeneration and growth of natural tissues [4–6]. In the context of bone TE, it involves building three-dimensional (3D) structures with controlled porosity to create a supportive environment for bone tissue growth. These structures are commonly known as bone scaffolds. TE, driven by recent advancements in material and fabrication processes, has become a highly viable approach for treating CSBD [1]. It is generally desirable to build scaffolds from biodegradable materials. Biodegradability allows the removal of the scaffold material from the patient's body when it completes its intended purpose without needing post-surgical procedures. The scaffolds and fabrication processes are expected to satisfy critical requirements to stimulate bone remodeling at the implant site. In upcoming sections, detailed discussions on selected fabrication processes and materials used in bone TE are presented. The reader is encouraged to refer to other reviews on manufacturing bone scaffolds for additional information not covered here, including recent reviews emphasizing 'bio-printing' with cellular elements [36, 43–50]. This review covers the potential application of electrospinning

in mineralized TE [1, 51, 52]. A survey of different manufacturing processes and the range of materials used specifically for bone scaffolds are provided.

2.3 Scaffolds for Bone Tissue Engineering

A *bone scaffold* is a temporary mechanical structure designed to mimic the ECM of bone tissues. Its primary objective is to provide an environment wherein bone remodeling can occur with minimal complications [53]. For a scaffold implant to be successful, *mesenchymal stem cells (MSC)* must first come in contact and adhere to its surfaces. Depending on availability at the implant site, the MSCs may need to be introduced to the implant site from external sources [4, 54]. Once this initial stage is completed, the bone grows at the scaffold's outer surfaces with an inward mineralization gradient [24]. The success of this process depends on several critical scaffold parameters such as surface roughness, internal architecture, porosity levels, material chemical composition, and biocompatibility. The ideal scaffold is characterized by superior mechanical strength, osteoconductivity, and biocompatibility. It has optimally sized and interconnected porous spaces to house the newly formed bone cells. The pore interconnections are vital because they serve as routes for blood vessels and transportation of nutrients and metabolic wastes [7]. Biodegradable materials are preferred in scaffold fabrication to minimize any residual synthetic matrix at the regeneration site. Cell attachment and inward growth are determined mainly by the scaffold's surface roughness and internal geometry, respectively. The porosity level in a scaffold directly impacts the bone remodeling process as it dictates the amount of surface area available for cell-scaffold interactions [55]. High porosity levels mean a higher internal surface area-to-volume ratio. Generally, the higher this ratio, the better the cell attachment and the bone growth characteristics [56]. However, there is an inverse correlation between the level of porosity inside a scaffold and its mechanical properties [53]. Excessive porosity leaves little material in the construct and adversely affects the mechanical performance of the scaffold. This poses a risk of structural collapse if the implant is exposed to external loads. Therefore, optimizing pore sizes is essential to balance structural and cell activity requirements.

Pore size requirements for a scaffold depend on the type of bone tissue for which the

scaffold is intended. Generally, smaller pore sizes ($< 75\,\mu m$) are found to be suitable for the formation of fibrous tissues. In comparison, intermediate (75–100 µm) and larger pore sizes ($200\,\mu m$) are more suited for unmineralized and fully mineralized bone formation, respectively. Bone formation, for example, requires a minimum of 100–150 µm pore sizes, while the requirement for vascularization is more than $300\,\mu m$ [24, 25]. Generally, pore sizes in the 50–1000 µm range are recommended for cell growth and complete recovery [55, 57].

Pore connectivity is another crucial geometrical parameter. Pore interconnections are channels that connect neighboring pores and function as transportation routes. When the scaffold degrades at the end of its useful life, the efficiency of its removal process is dictated by the sizes of these interconnection channels [56]. Channel sizes in the range of 15–50 µm are generally suggested for enhancing inter-porous flow properties [58, 59]. Among the many attributes of a suitable scaffold, the following stand out as the most crucial [25, 53, 60]: 1) surface permeability to allow the transfer of biological fluids at the boundary, 2) surface roughness to promote cell attachment and invasion critical for osteoconductivity, 3) biocompatibility to minimize the risks of local toxicity and undesired immune/inflammatory reactions, 4) biodegradability and removal of degradation products by the metabolic processes [61, 62], 5) internal architecture and porosity distribution to facilitate the scaffold invasion with cells for tissue induction, and 6) matching mechanical properties to that of neighboring tissues at the implant site [7, 36]. These requirements underscore the importance of understanding the relationships among the scaffold material, internal geometry, and other process parameters and their impact on clinical outcomes.

2.4 Scaffold Materials for Fabrication Processes

Several materials and fabrication methods meet the scaffold requirements. However, not all materials and fabrication methods are compatible, and the right combination should be selected when designing a process for scaffold fabrication. The choice of the fabrication method also dictates whether drugs, growth factors, and other biological ingredients could be directly impregnated into the base material during the fabrication process. The additives are released at the implant site when the

scaffold material undergoes degradation. The effectiveness of the drug release rate is influenced by, among other factors, the ability of the fabrication process to control the drug concentration and spatial distribution. High-temperature fabrication methods, for example, could thermally degrade these additives and are not best suited for such applications [13, 63].

One widely used material for the preparation of bone scaffolds is hydroxyapatite (HA), with a chemical formula $Ca_{10}(PO_4)_6(OH)_2$ and a Ca:P ratio of 1.67. It is naturally available at the bone site, has high mechanical strength, and can undergo cellular degradation without obvious toxic byproducts. Beyond the HA, other related Ca/P-based ceramics have been employed in making bone scaffolds, including brushite (DCPD; $CaHPO_4\,2\,H_2O$ with a Ca/P ratio of 1.00), octacalcium phosphate (OCP; $Ca_8(HPO_4)_2(PO_4)_4\,5\,H_2O$ with Ca/P ratio of 1.33), α-tricalcium phosphate (α-TCP; $Ca_3(PO_4)_2$ with Ca/P ratio of 1.5), β-tricalcium phosphate (β-TCP; $Ca_3(PO_4)_2$ with Ca/P ratio of 1.5) and tetracalcium phosphate ($Ca_4(PO_4)_2O$ with Ca/P ratio of 2.0) [64]. They have varying degrees of osteoinductive and osteoconductive properties. Their biodegradable rates could be tuned by carefully optimizing the Ca:P molar ratio in the compound.

To enhance tissue integration and overcome the shortcomings of the pure form of HA, one can employ materials that possess similar attributes to the constituent ingredients of the natural bone (HA minerals and collagens) [7, 25, 65]. This observation has brought to focus the use of artificially synthesized nanohydroxyapatite (nHA) composites, with glasses and other minerals added as fillers [4, 41, 66]. This approach improved the mechanical properties of the scaffold material and allowed customizing the materials to meet specific functional and geometrical requirements. For example, a study conducted by Goncharuk *et al.* [67] demonstrated improvement in the surface-to-volume ratio by synthesizing an nHA-polysaccharides (PS) composite using a liquid-phase two-step solution. The nHA nanoparticles were first synthesized from a wet reaction between $Ca(OH)_2$ and phosphoric acid according to:

$$10\,Ca(OH)_2 + 6\,H_3PO_4 \longrightarrow Ca_{10}(PO_4)_6(OH)_2 + 18\,H_2O. \tag{2.1}$$

The nHA/PS composite was synthesized in the second stage by mixing the nHA

suspensions from the first stage in a PS solution. They reported improvements in the structural, morphological, and thermal behaviors of the resulting composite material. For example, the specific surface area, an indirect measurement of surface-to-volume ratio, was reported to vary between 0.3–43 m^2/g to 49–82 m^2/g for a 1:1 and 4:1 ratio of nHA/PS, respectively. Using a similar approach, Shakir *et al.* [68] first synthesized the nHA particles from the reaction between calcium nitrate tetrahydrate ($Ca(NO_3)_2\,4\,H_2O$) and diammonium phosphate ($(NH_4)_2HPO_4$) according to the following chemical reactions [69–71]:

$$10\,Ca(NO_3)_2\,4\,H_2O + 6\,(NH_4)_2HPO_4 + 8\,NH_3 \longrightarrow Ca_{10}(PO_4)_6(OH)_2 + 20\,NH_4NO_3 + 38\,H_2O \quad (2.2)$$

$$10\,Ca(No_3)_2\,4\,H_2O + 6\,H_3PO_4 + 20\,NH_4OH \longrightarrow Ca_{10}(PO_4)_6(OH)_2 + 20\,NH_4(NO_3) + 58\,H_2O. \quad (2.3)$$

The nHA was co-precipitated with the PS chitosan (CS) and polyethylene glycol (PEG) to synthesize the nHA/CS/PEG nanocomposite. They reported improved scaffold performance in mechanical properties, surface morphology, biotoxicity, and crystallinity. A comparison between nHA/CS and nHA/CS/PEG showed that adding PEG to nHA/CS/PEG improved the *in vitro* bioactivity. Several other studies [4, 40, 54, 64, 68, 70–75], using similar chemical reactions to synthesize nHA, have reported improvements in HA-based material properties using similar approaches.

Natural and synthetic biodegradable polymers and their combinations have been tested for bone scaffold applications. While the naturally occurring polymers in Table 2.1 tend to possess excellent biocompatibility and physiological degradation characteristics, they particularly suffer from weak mechanical strength. These materials are compatible with select fabrication processes, including hydrogelation, cryogelation, fiber bonding, and photopolymerization. The synthetic polymers (Table 2.2) offer superior mechanical attributes relative to the natural polymers but suffer inferior cell attachment properties. Their degradation rate has also been directly linked to the creation of an acidic environment.

Table 2.1: Representative natural bio-polymers in bone tissue engineering [76–81]

Source	**Polymer**	**Basic properties**	**FDA approved**
Proteins (extracted from animal and plant-based biomass)	Collagen	• Biocompatible and supportive of cell attachment • Can be created as a spongy hydrogel • Supportive of osteoblast differentiation • Biodegradability with no acidic byproduct • Poor mechanical properties • Can be modified by ceramics and bioglasses • Can be made into composite polymer	Yes
	Fibrin	• Hydrogel • Biodegradable and good cell interaction	Yes
	Gelatin	• Can be created as a spongy hydrogel • Biodegradable and bioresorbable	Yes
	Silk sericin	• Successful cell attachment • Biodegradable • Used for wound dressing [82] • High tensile strength • Successful cell attachment	Yes
	Keratin	• Osteoconductive • Exhibits excellent cell adhesion properties • Supports vascularization • Slow degradation rate	N.A.
Polysaccharides (extracted from animal and plant-based biomass)	Alginate	• Hydrophilic • Can create hydrogel scaffolds • Biodegradable with no toxic byproduct	Yes
	Chitosan	• Has the antibacterial property • Can create hydrogel scaffolds • Composite scaffold with ceramic fillers • Biodegradable	Yes

Continued on next page ...

Source	Polymer	Basic properties	FDA approved
	Hyaluronic acid	• Major component of cartilage ECM • Leads to highly hydrated (hydrogel) scaffolds • Could be made into an injectable gel	Yes
Biopolyesters *Poly(hydroxyalkanoate)s (PHA)s* (derived from microorganisms)	Poly(3-hydroxybutyrate) P(3HB)	• Possesses excellent biocompatibility • Brittle in nature, but mechanical properties can be improved by mixing with other PHA monomers or ceramics [83–85]	N.A.
	poly(3-hydroxybutyrate-co-3-hydroxyvalerate) (PHBV)	• Wide range of physiochemical properties can be tailored as a result of variation in chemical structure within the PHA group [86] • Thermally processable	N.A.
	poly(3-hydroxybutyrateco-3-hydroxyhexanoate) (PHBHHx)	• Significantly better cell attachment and proliferation properties (about 40% after 10 days) than P(3HB) [50, 87]	N.A.
	poly(3-hydroxybutyrate-co-3-hydroxyvalerate-co-3-hydroxyhexanoate) (PHBVHHx)	• Relatively newer material • Rougher texture and more hydrophobic than PHBHHx • Better cell adhesion and osteogenic differentiation (66% after 72 h) than PHBHHx [88]	N.A.

Table 2.2: Representative synthetic bio-polymers in bone tissue engineering. (Adapted from [89])

Synthetic polymer	**Degradation rate** (months)	**Modulus of Elasticity** (GPa)	**Properties**	**FDA approved**
Poly(caprolactone) (PCL)	> 24	0.4–0.6	• High permeability to drugs [82] • Less acidic due to slow degradation	Yes
Poly(lactide-co-glycolic) acid (PLGA)	3–6	1.4–2.8	• Optimal composition for optimal degradation • Acidic degradation byproducts	Yes
Poly(*L*-lactic) acid (PLLA)	> 24	1.3–1.3	• Degrades into non-toxic byproduct [82] • Inflammation due to the crystallinity of degraded byproduct • Injection moldable	Yes
Poly(*D,L*-lactide) acid (PDLLA)	12–16	1.9–2.4		
Poly(propylene-fumarate) (PPF)	> 24	2–3		

Creating composite materials to address challenges experienced by one-dimensional materials has yielded more optimal materials and functionalized scaffolds. Generally, ceramic and bioglass minerals are added to the natural and synthetic polymers to create scaffolds with enhanced mechanical and biological performance. The presence of ceramics such as HA and TCP in the composites improves the scaffold's compressive strength, degradability rate, and osteogenic capacity. Meanwhile, the polymers maintain good elastic strength and provide a cross-linking mechanism. Composites such as TCP/TTCP/Polymer, β-TCP/bioglass, HA/Starch, HA, CPP/PVA, HA/gelatin, and PCL/HA are frequently reported in use [24, 90, 91]

2.5 Fabrication Processes

Most fabrication methods, particularly those under additive manufacturing (AM), rely heavily on a digital representation of the target tissue for processing materials into bone scaffolds. Acquiring and processing this data by itself has become the focus of numerous studies as the importance of the internal micro-structures of bone scaffolds became evident [92–94]. The collected data may be used to model new scaffolds or to evaluate the state of deployed scaffold implants [95]. To design a new scaffold, macro-structural data (data pertaining to the external shape of the targeted tissue) and micro-architectural data (internal pores and interconnective channels) from the target bone tissue are needed. Several well-established bioimaging techniques could be employed to achieve this. While qualitative computed tomography (CT) is generally used for collecting macrostructural data, high-resolution computed tomography (hrCT) and high-resolution magnetic resonance (hrMR) produce good results for micro-architectural data collection *in vivo*.

On the other hand, micro-computed tomography (μCT) and micromagnetic resonance (μMR) are frequently used for *in vitro* evaluations [96]. Even though these systems work based on 2D scanning, the 3D representation of the tissue geometry is constructed by volume rendering from the 2D data [97, 98]. The resulting geometry is commonly stored and communicated in stereolithography (STL) file format. Since STL represents 3D models by approximating the enveloping surfaces of the solids by stitched triangular facets, it slightly loses accuracy when approximating

curved surfaces. Challenges related to data redundancy and inability to carry other manufacturing-related information have been reported [99, 100]. Even though several different file formats have been developed to address these shortcomings, STL is still widely accepted as an industry standard.

Depending on the fabrication method involved, the STL file is further processed to generate fabrication method-specific data. For example, this may be 2D nozzle paths for powder and extrusion-based printers and projectable cross-sectional images for vat photopolymerization-based fabrication methods. Several techniques have been proposed to enhance the mechanical and cell-guiding properties of scaffolds based on the optimization of the internal architecture of the scaffold. While most rely on patterning and filling the interior of the scaffold with microcellular cavities of primitive shapes [101–104], complex and spatially-graded internal architectures are also getting traction [105–108].

The internal geometry is an important parameter that dictates room available for osteogenic activities. There is a growing interest in replicating the internal geometry of the natural bone tissue in the scaffolds [109]. However, this practice poses challenges in acquiring the data in the first place and reproducing it with existing technology cost-effectively. Its effectiveness in promoting bone growth is also debatable as the scanned data represents an already matured bone tissue, which is not necessarily a geometry that promotes the growth of new bone cells into this matured state.

Numerous fabrication processes have been developed to take advantage of bone's regenerative potential, and below are summarized the most popular current fabrication techniques.

2.5.1 Gas foaming

Gas foaming (GF) is the process of creating porous structures by forming bubbles or passing gases inside a material blend of amorphous consistency [42, 110]. The process, depicted in Fig. 2.2, starts with mixing the substrate, a foaming agent, and a binder into a jelly-like or amorphous consistency. The mixture is then molded into the shape

of the target scaffold using a convenient method and allowed to solidify partially. The material is immersed in a solution bath that chemically reacts with the embedded foaming agent. As the chemical reaction synthesizes gasses, the erosion caused by the escaping gas bubbles inside the pre-molded body creates the required porous internal geometries. Alternatively, inert or non-reactive gases such as N_2 and CO_2 supplied from external sources are forced to pass through the substrate to create the required internal porous structures [56]. As summarized in Table 2.3, this method has been extensively used to investigate different materials for bone scaffold applications.

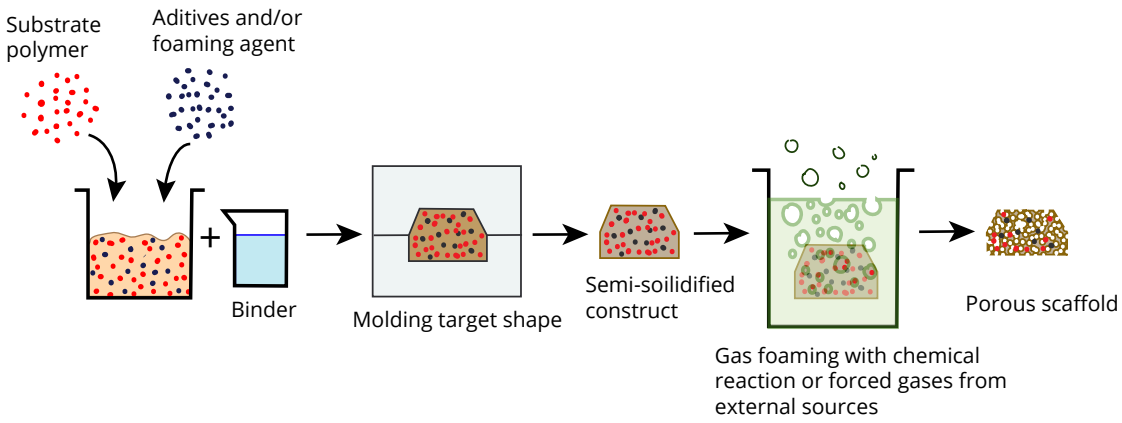


Figure 2.2: Schematic diagram of a typical gas foaming process. Adapted from [56]

The pore sizes are generally controlled by controlling the mixing ratio of the constituent materials and foaming agents. The consensus is that the more gas the process generates, the higher the porosity level. Depending on the type and mixing ratio of the substrate, foaming agents, reverse templates, and solvents, pore size in the range of 40–800 µm can be achieved. Table 2.3 summarizes materials used in two different gas foaming processes.

Table 2.3: Materials used in gas foaming

Base material	**Gas source**	**Foaming reaction**	**Solvent**	**Pores (µm)**	**Process attributes**	**Ref.**
PLGA	Chemical reaction	NH_4HCO_3 (salt) + aq. citric acid	N.A.	200–500	• Fast, cost-effective • Pore inter-connections not always guaranteed	[110, 111]
PCL and thermo-plastic gelatin (TG) 40%PCL + 60%TG	Forced flow of N_2 and CO_2 mix	External gas supply	water (to dissolve TG)	40	• Fine pore size • Improved connectivity	[56]
40% low MW PCL + 60% high MW PCL + HA	Super critical CO_2	External gas supply	Solvent used for polymer extraction: water	200–600	• Biocompatible • Mechanically strong	[112]
PCL + drugs	Supercritical CO_2	External gas supply	Drug solvent	75–250	• Morphology and crystallinity depend on drug and solvent concentration. • High solvent gave high porosity	[113]
30%PCL/70% NaCl	CO_2 and N_2 mixture	External gas supply	Waster to dissolve NaCl	300–500	• Reverse templating used with NaCl	[56]
PLA	CO_2	External gas supply	None	Two-stage: 100–800 1–10	• A hierarchical process; fused deposition of PLA first, followed by gas foaming	[114]

The main drawback of this process is its inability to create fully interconnected pores. Different methods have been proposed to resolve this limitation. The use of secondary-stage foaming agents is one possibility. These agents, also known as residual opening phases, help create more porosity by either triggering a secondary foaming reaction at a later stage or initiating mechanical deformations that create additional pores and channels. Scaffold porosities can be further improved by carefully selecting the residual phase to serve as a reverse template. In reverse templating (sometimes known as selective polymer extraction), one or more constituent materials in the mix are removed by dissolving in a solvent to create additional voids in the structure [56]. Salerno *et al.* [56, 110, 115] demonstrated this by adding water-soluble residual opening phases such as NaCl and thermoplastic gelatin into the substrate [115]. After serving the primary purpose of residual phasing, the salt and the thermoplastic gelatin were dissolved away, leaving behind more voids in the gas-foamed structure and effectively increasing the porosity and interconnections [56].

Another method proposed by Zhou *et al.* [114] attempted to utilize the combined strengths of AM and gas foaming to deliver scaffolds with improved porosity. The proposed method is based on a hierarchal approach in which the macro-level details of the scaffold were first fabricated out of PLA using extrusion-based 3D printing techniques. The internal pore sizes achieved by this initial stage were reported to be in the 100–800 µm range. The printed PLA part was later foamed using CO_2 as a foaming agent, resulting in a final porosity of 1–10 µm. The gas foaming improved the porosity significantly and allowed greater control over the geometry of the scaffold at the macro level.

Nanoparticles are sometimes added to the substrate mix to enhance the mechanical and clinical performances of the scaffold. A study by Torres *et al.* [7] showed an improvement in these two areas by adding TiO_2 into a PLGA substrate. In addition to serving as fillers and improving the mechanical properties of the scaffold, the added nanoparticles enhanced the cell attachment and the subsequent inward proliferation during clinical trials. Because of its excellent biocompatibility and osteoconductive characteristics, HA has seen application as a nanoparticle additive in gas-foamed substrates [42, 112]. Drugs and growth factors may also be added to the substrate to

aid tissue induction. Salerno *et al.* [113] recently investigated the effects of three drugs (5-fluorouracil, nicotinamide, and triflusal) on scaffold morphology and their release rate. The group used supercritical CO_2 foaming to fabricate the scaffold out of PCL-drug mixes. A CO_2-inert organic solvent was used to dissolve the drugs during the substrate preparation stages. They reported that the drug and solvent concentration affected crystallinity, morphology, and pore size. Varying results were also observed among the drugs considered.

The other drawback of gas foaming is its unsuitability for creating intricately shaped scaffolds. Gas foaming of such geometries requires a 'negative scaffold' to be fabricated first and used as a mold in the actual foaming process later, which may not be a cost-effective approach. Gas foaming has the crucial advantage of lower cost and faster processing time than AM. However, the internal geometries are relatively harder to control; residual foaming agent and solvent toxicity pose a risk; and the process, in its traditional form, cannot achieve predefined internal geometries [56]. As demonstrated by Zhou [114], the latter can be addressed by merging the two processes.

Pure polymers such as PLA, PGA, and PCL have rheological properties that make them exceptionally suitable for gas foaming. As a result, they have been extensively used in producing scaffolds with attractive morphologies [116, 117]. From a biological performance perspective, the molecular weight of the polymers, their osteoconductivity, and the nano-scale surface roughness play significant roles. They could affect the initial cell attachment, seeding (if *ex vivo*), proliferation, differentiation, and mineralization stages. The use of high molecular weight polymers has been associated with inferior geometrical properties such as smaller pore sizes, porosity, and interconnectivity due to the high viscosity that characterizes these polymers. High viscosity generally leads to smaller and discrete foam cells, which is detrimental to the resulting scaffolds' biological cell attachment and growth behaviors [112]. The ability of polymers to support osteoconduction is usually improved by adding or mixing ceramic fillers to form polymer-ceramic composite scaffolds. Compared to pure polymers, ceramic composites' benefits were evident in their cellular performance. Seeded with human osteoblast cells under similar conditions, adding a small amount (1–5%) of

HA and TCP minerals, for example, improved cell attachment and osteoblastic activities [42, 118]. Additional micro-surface modification using gas plasma treatment can further improve the permeability and wettability of gas-foamed scaffold surfaces and enhance their cell adhesion and seeding performances [119]. Other fabrication methods, such as AM, also use this surface roughness and permeability modification methods.

It appears that the initial stages of the bone remodeling process have been, to some degree, successfully replicated *in vitro* on gas-foamed scaffolds. Varying results in cell attachment, seeding, proliferation, differentiation, and mineralization have been reported in response to material and process parameter factors.

2.5.2 Cryotropic Gelation

Cryotropic gelation, also known as cryogelation, is the process of creating microporous structures by freezing a polymeric solution suspended in water (or other solvents) to a sub-freezing temperature and allowing a phase separation to occur between the solvent and the precursor solution (Fig. 2.3) [120]. At these temperatures, the water crystalizes, and the precursor solution goes through a polymerization phase resulting in permanent crosslinks between its molecules. The porous structure is created when the frozen water crystals trapped inside the body of the polymerized gel are thawed and extracted upon unfreezing [121–123].

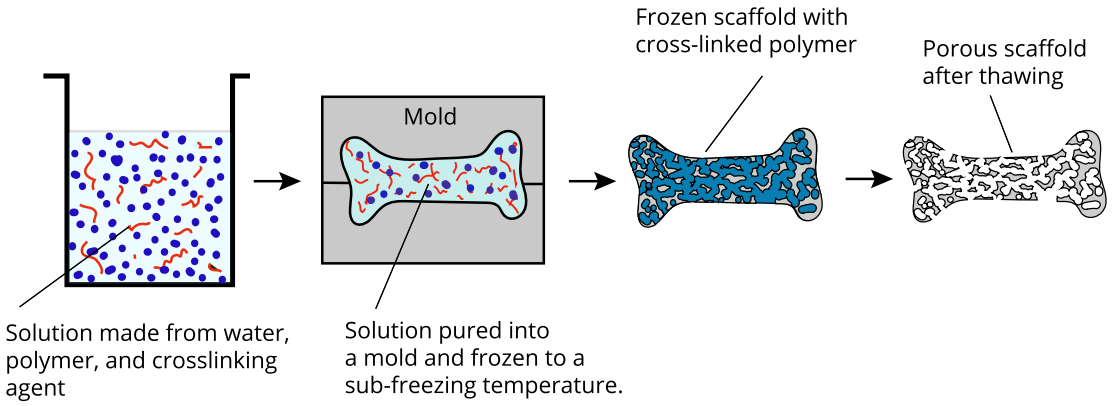


Figure 2.3: Working principle of cryotropic gelation (Adapted from [121])

The gentle process parameters and availability of a range of biocompatible polymers

suitable for cryogelation have made this process another popular choice, especially for incorporating growth factors in TE scaffolds [22, 124–128]. HA nanoparticles and bio-additives are often added to the precursor solution to improve the mechanical and clinical properties of the scaffolds. For example, a study conducted by Salgado *et al.* [129] reported improvement in cell adhesion, differentiation, and biodegradation characteristics when a composite scaffold made from a collagen/HA mixture was crosslinked at $-10\,^{\circ}\mathrm{C}$ and seeded with cells both *in vivo* and *in vitro*. Another study [130] used a similar method to study the combined effects of an osteogenic protein (BMP-2), a bioglass additive, and a drug (bisphosphonate). The drug was included in the study to prevent resorption caused by the protein as a side effect. The precursor solution was formulated by mixing the protein and the drug with silk-fibroin, chitosan, agarose, and nHA. Two types of scaffolds were fabricated by crosslinking this solution at $-80\,^{\circ}\mathrm{C}$ with and without the bioglass additive to evaluate the effects of bioglass additives. After seeding the scaffolds with myoblast (C2C12) cells and MSC *in vitro* and observing the cell activities for 60 days, the team reported gradual and sustained cell proliferation and differentiation. The bioglass was found to have no significant effect on the drug release rate. One way of improving the mechanical properties is by employing multi-stage crosslinking in which the pores of a scaffold fabricated by cryogelation are filled with precursor polymer solutions and crosslinked one or more times. This approach produced a fibroin-glycol scaffold in a two-stage crosslinking operation [131]. Significant improvement in the mechanical properties of the scaffold was reported ($E = 66$–126 MPa, 90% compression under 87–240 MPa applied stress). Although the material added in the additional crosslinking stages was credited for improving strength, its impact on the pore sizes and interconnection quality is yet to be determined.

The duration of the crosslinking stage impacts the scaffold's morphology directly. In general, the longer the crosslinking time, the higher the level of porosity and the thinner the wall thickness separating these pores become [121].

One of the advantages of cryogelation is that it allows control of the porosity size by carefully selecting the solvent and precursor solution ratio. Impregnating biological factors into the mix is easy, and the low temperatures in the process eliminate the

risk of thermal degradation of drugs and growth factors. Porosity and mechanical properties of the scaffold construct could also be altered by adding composite fillers, fibers, and other polymers into the solution mix [132]. For example, Kai *et al.* [133] have reported that adding electrospun PCL fibers into a gelatin hydrogel solution increased the elastic modulus of the resulting construct from 3.29 ± 1.02 kPa to 20.30 ± 1.79 kPa. Compared to most AM methods, the limitations of this process arise from a lack of mechanisms to precisely define and control the scaffold's internal and external geometries. Although the average pore size is controlled by altering the material composition, freezing temperatures, crosslinking time, and adjusting other process parameters, achieving a pre-determined size distribution remains challenging. This is at odds with the current trend in bone scaffold engineering, which requires precisely defined and spatially distributed internal architectures.

Like gas foaming, pore interconnections are only sometimes guaranteed in cryogelation because the porosity is randomly created as a byproduct of the chemical interaction between the molecules of the precursor solutions. The porosity level could be improved by using different material additives and increasing the crosslinking time (Table 2.4). The other geometrical limitation of this method is that it requires additional procedures and possibly molds to create the outer shape of the scaffold. Despite these limitations, cryogelation remains an invaluable tool for the initial evaluation of different materials and growth factors [134].

Table 2.4: Commonly used materials in cryogelation

Material	**Crosslinker**	**Pore size** (µm)	**Comment**	**Reference**
Methacrylamide modified gelatin		135 ± 10	Ultrasound and electromagnetic simulation resulted in increased production of ECM	[135, 136]
N(2-Hydroxypropyl) methacrylamide (HPMA) Copolymer of HPMA and methacryloyl-GGGRGDS-OH	Poly(ethylene glycol) dimethacrylate	50-300	Concentration of crosslinker affected elasticity of biomimetic hydrogel	[137]
Methacrylated chondroitin sulfate (MeCS), Methacrylated hyaluronic acid (MeHA)	Poly(ethylene glycol) diacrylates (PEGDA)	63 ± 19 59 ± 18		[125]
Chitosan gelatin (CG) with HA, CG, and BC as fillers		30	Improved rate of mineralization due to ceramic fillers	[138, 139]
HEMA-Lactate-Dextran		NA	Higher level of vascularization with a cell-seeded scaffold.	[124, 140]
PEG with graphene oxide (GO) as a filler	Ammonium persulfate Tetramethylethylenediamine	NA	Addition of GO stimulated osteogenic differentiation	[141]
Gelatin/HA	Glutaraldehyde	200/50/50 (10/30/50%HA)		[142]
Gelatin/PCL	Methacrylamide	156–359	Hierarchical scaffold (3D printed PCL later cryogelled for strength)	[127]
PEG-HEMA		50	Small amount of methacrylated poly(g-glutamic acid) improved mineralization, but a not higher percentage.	[143]
N-vinyl-2-pyrrolidone (NVP)			Synthetic biocompatible monomer	[139]
Silk fibroin (SF)		19	Natural polymer	[139]
Polyvinyl alcohol-tetraethylorthosilicate-alginate-calcium oxide (PTAC)			Material possess good osteoinductive and osteoconductive properties	[144]
Heat-treated polyacrylonitrile with succinonitrile porogen		12–20	HT-PAN resulted in acceptable scaffold morphology	[145]

With the design of cryogel scaffolds, *in vivo* cell penetration into the scaffolds could be realized without difficulty. A comparative study between a blank scaffold and an MSC-seeded scaffold showed no significant difference in cell viability, proliferation, differentiation, and mineralization characteristics. However, accelerated vascularization was observed in the MSC-seeded scaffolds. A separate study has shown, consistent with other fabrication methods, improvement in osteoblastic activities in the presence of ceramic fillers [138, 141]. The degree of crosslinking, however, was negatively impacted by the presence of ceramic fillers. For example, adding 10, 30, and 50% HA to gelatin ratio resulted in a composite cryogel with 200, 50, and 55 µm pore sizes, respectively, accompanied by order of magnitude increases in mass densities [142]. The ceramic filler's neutrality to the crosslinking reaction and its tendency to disrupt the links physically can explain this observation. In another case, improved mineralization was observed for up to a certain percentage when mPGA was added to PEG-HEMA scaffolds [143].

External physical stimuli, including electrical current, ultrasound, and electromagnetic sources, have also been used in some cases to stimulate cell migration and osteogenetic differentiation *in vitro* [135, 144, 146]. Comparison between stimulated and control scaffolds showed increased secretion of ECM components in the stimulated scaffolds. Higher levels of osteocalcin, osteopontin, and Type-I collagens were reported in response to the stimulation (2.9-fold, 2.4-fold, and 1.5-fold, respectively).

The combined use of multiple manufacturing methods to address some of the limitations of cryogelation has been gaining much attention recently. Van Rie *et al.* [127] produced a scaffold with improved mechanical properties using 3D printing and cryogelation processes. In their design, an initial scaffold was 3D printed out of PCL with sparse resolution and later flooded with gelatin cryogel solution. Upon crosslinking, the scaffold was characterized by higher mechanical strength (80 MPa) due to the stronger PCL core while still exhibiting micropores created by the cryogelation. However, the pore sizes in the 156–359 µm range were suboptimal compared to scaffolds made by cryogelation alone (50–75 µm).

2.5.3 Additive Manufacturing

Additive manufacturing (AM) or solid free-form fabrication (SFF) is a collective term used to describe a group of fabrication methods in which 3D objects are constructed layer-by-layer in an incremental manner [147–150]. The process starts by creating a computer-aided design (CAD) model for the target scaffold. While some simpler models could be manually constructed using commercial CAD tools, complex shapes, such as bone ECM, are problematic to model manually and may require scanning and digitizing techniques. Computed tomography (CT) and magnetic resonance imaging (MRI) techniques have been proven to be valuable tools in this regard [25, 151]. An intermediate software usually converts the generated 3D model to formats readable by most AM equipment. *Stereolithography* (STL), *object* (OBJ), and *additive manufacturing file* (AMF) are some examples of widely used file formats. The 3D model is digitally sliced into multiple layers represented by their 2D profiles [152]. The type of information contained within the 2D profiles varies from one AM method to another. In fused deposition modeling (FDM), for example, the data may contain the paths to be followed by the nozzles and the coordinates at which the nozzles start and stop extruding [4, 40]. On the other hand, 2D cross-sectional data generated for a vat photopolymerization may contain mainly projectable images of the cross-sections.

AM has unique capabilities vital to producing bone scaffolds with precisely predefined external and internal architectures. This method is the best for controlling the general shapes of printed scaffolds on the macro level. In addition, limited success has been reported in producing scaffolds with internal gradient architecture. Spatial variation of internal architecture could lead to different vascularization profiles, leading to the growth of different types of tissues (bone, cartilage, and vascular) within the same scaffold [153, 154]. Gradient internal architectures also play critical roles in enhancing the mechanical strength of the scaffold [155]. However, despite these advantages, AM methods are best suited only for a few scaffold materials. Factors such as material compatibility, printing resolution, cost, and processing speeds must be considered when selecting a fabrication method for a specific application.

Binder Jetting

Binder jetting is one of the most widely investigated AM methods for bone TE. A chemical interaction between a powder scaffold material and a liquid binding agent binds the loose powder particles into a solid entity [25, 41, 43, 156–158]. A precisely controlled binder delivery system selectively sprays the binder onto regions of the powder bed where binding is required (Fig. 2.4). Considering the volumetric shrinkage of the material due to wetting, the thickness of this powder layer is selected to match the slice thickness established in the preprocessing software [90]. When wet by the liquid binder, the material goes through several transformation stages that ultimately solidify the activated regions. A detailed discussion on these stages and powder wetting mechanism can be found in Reference [159]. While the wetted region hardens to become the solid cross-sectional segment of the printed construct, the dry regions remain in their original powder form serving as support structures for overhanding features in upcoming layers. When the binder spraying is completed for the current layer, the printing platform is lowered, and a new layer of fresh powder material is added at the top of the bed. These steps are repeated until the entire 3D object is entirely constructed. The loose powder around the construct is removed at the end of the printing process to free the part [4]. The process of removing the unbound powder is commonly known as *de-powdering*.

The powder particle sizes and shapes affect the powder deposition characteristics of the material and the de-powdering process. Generally, smaller particle sizes (<5 µm) with rugged edges are characterized by poor particle flow. They tend to agglomerate during wetting under the effects of Van der Waals forces [160, 161]. Rounded and large particles (>20 µm), on the other hand, have less internal friction and a tendency to flow and spread well [157, 162]. Several studies have proposed different methods for controlling scaffolds' mechanical properties and internal architectures based on this observation [163]. This behavior was also independently demonstrated by Utela *et al.* [157] and Zhou *et al.* [159]. They manipulated the inter-particle spaces between contacting powder particles to achieve targetted porosity and interconnective channel profiles. They reported that powder granularity directly influenced the size and quality of these features. The wetting and subsequent post-processing procedures also

affected the surface finish of the scaffold.

Butscher *et al.* [90] proposed a two-part scaffold design to enhance the de-powdering characteristics based on the powder particle profiles. The first layer—an outside solid shell constructed with large pore sizes and superior mechanical strength—was designed to enclose and house free-flowing powder particles inside. The study evaluated the effect of different filler particle profiles (rounded/rugged) on the de-powdering characteristics. The result showed enhanced de-powdering compared to scaffolds printed with solid cores. However, a significant reduction in compressive strength (14–18%) was also reported due to a reduction in material density.

Its accuracy, flexibility, and suitability for processing a wide range of materials and binders at low cost have made binder jetting one of the most attractive methods for scaffold engineering. Materials compatible with this method include metals, ceramics, polymers, and composites [157]. However, some clinical requirements have limited the number of compatible material-binder combinations suitable for bone scaffold applications [159]. While some materials and binders have been found to contain toxic chemicals, others release toxic gases when exposed to higher temperatures during the post-processing stages. Because of their close similarity to the mineral composition of the natural bone tissue, variations of Ca/P have been widely investigated by this method (Table 2.5).

One major drawback of binder jetting concerns the impregnation of drugs and growth factors directly into the matrix [190]. The scaffold constructs directly coming out of this process (also known as green parts) are characterized by low mechanical strength. These parts are usually sintered at high temperatures to improve their mechanical characteristics, [4, 31, 54, 191, 192]. Although this practice effectively enhances the mechanical strength of the constructs, it could thermally degrade drugs and growth factors.

The binder droplets in binder jetting are also known to cause impact craters on the powder layer upon impact, imposing an additional limitation on the print resolution. Nevertheless, this fabrication method generally yields a print resolution range of 40–500 µm, which fulfills the requirement for most bone scaffold applications [25, 171].

Table 2.5: Materials and binders used in binder jetting for bone TE

Material Name	**Formula**	**Binder**	**Ref.**
Hydroxyapatite (HA)	$Ca_{10}(PO_4)_6(OH)_2$	(10%) H_3PO_4, schelofix polymer	[164, 165]
Tricalcium phosphate (TCP) - (α/β)	$Ca_3(PO_4)_2$	(5-30%) H_3PO_4, water	[166–170]
Tetra calcium phosphate (TTCP)	$Ca_4(PO_4)_2O$	(10%) H_3PO_4, (25%) citric acid	[25, 37, 65, 171]
Dicalcium phosphate dihydrate (DCPD)	$CaHPO_4\,2\,H_2O$	(10%) H_3PO_4	[172–174]
Dicalcium phosphate anhydrous (DCPA)	CaHPO4	(10%) PVA	[175]
Octacalcium phosphate (OCP)	$Ca_8H_2(PO_4)_6\,5\,H_2O$	(5-30%) H_3PO_4	[172, 176–178]
Calcium pyrophosphate (CPP)	$Ca_2P_2O_7$	(5-30%) H_3PO_4	[179, 180]
Calcium sulphate (CS)	$CaSO_4\,0.5\,H_2O$	water, (25%) citric acid	[159, 181–183]
Calcium phosphate + HA	α-$Ca_3(PO_4)_2$ + $Ca_{10}(PO_4)_6(OH)_2$	H_3PO_4, water-based	[156, 179, 184]
PLA	$(C_3H_4O_2)_n$	Chloroform	[185–187]
High density polyethylene (HDPE)	$-(CH_2-CH_2)_n-$	Maltodextrin + PVA	
Starch + dextran + gelatin	$(C_6H_{10}O_5)_n$	Distilled water	[188]
Titanium alloys	$Ti-6\,Al-4\,V$	Starch-based aqueous solution	[189]

Multi-material powder mixtures are also commonplace in this process. They can be used to take advantage of the strengths of the constituent materials. Generally, adding calcium-based ceramic minerals to metallic and polymer bases improves the osteoactivities of the scaffolds [171]. An excellent example of this approach is the work of Hong *et al.* [193], where the cell viability and cytocompatibility of a biodegradable Fe-Mn scaffold were analyzed in the presence of calcium in the mix using mouse pre-osteoblast MC3T3-E1. Indirect MTT viability assays on Fe-Mn and Fe-Mn-Ca scaffolds later revealed that the latter resulted in about 50% increased activity. Additional data also indicated that the Ca mineral increased the corrosion rate from 0.04 mm/year for the Fe-Mn scaffold to 0.07 mm/year for the Fe-Mc-Ca scaffold, suggesting improved biodegradability of the modified scaffold. In addition to modifying metallic materials, ceramic materials could also be used to modify other ceramics. For example, mixing HA and TCP into BCP has been reported to promote the mechanical strength of scaffolds [194]. In addition, the presence of HA and TCP enhances osteoconductivity and high biodegradability, respectively. Castilho *et al.* [195] reported that the TCP scaffolds alone were as good as the HA-control scaffolds in bioactivity and osteoactivity. However, the BCP had the additional benefit of improved biodegradability profile of the scaffold.

Powder Bed Fusion (PBF)

Powder bed fusion is a category of AM in which thermal energy sources, instead of binder solvents, are used to melt and fuse selected regions of a powder material bed. Depending on the material, the thermal energy source, and the level of heating involved, the manufacturing processes may be further categorized into (i) selective laser sintering, (ii) selective laser melting, (iii) direct metal laser sintering, and (iv) electron beam melting.

Selective Laser Sintering (SLS). In SLS (Fig. 2.5), a laser energy source is employed to raise the temperature of the material to just under the melting point. The high temperatures fuse the powder material particles without triggering an actual melting. In most cases, to reduce the laser power requirement and speed up the process, the sintering usually takes place inside a heated enclosed chamber [196]. The absence of melting plays a vital role in preserving the micro spaces between the powder particles. These inter-connected spaces comprise the required ECM-like porous structure inside the printed scaffold [99, 196, 197]. Since SLS relies on these inter-particle voids to create the micropores, careful optimization and selection of the grain sizes are of paramount importance [198–201]. Generally, the process is capable of producing constructs with an average macropore resolution of 400 µm and inter-particle spacing of 50 µm [6]. Layer thickness achieved by this process ranges from 50 to 200 µm [199]. The process follows similar steps outlined in the binder jetting AM methods for preparing and handling the powder material layers.

Suitable materials for SLS include PLLA, PCL, polyvinyl alcohol (PVA), polyether-ether-ketone (PEEK), and polyamides, such as Polyamides (PA), such as PA4,10, PA6,10, PA10,10, and PA11. Many other materials listed in Table 2.6 have also been investigated using this method [204–211]. Metallic and ceramic materials and other composite fillers could also be added to the polymer base to enhance scaffold characteristics. Duan *et al.* [212] and Liu *et al.* [213] tested this by adding HA and TCP fillers to a PCL substrate. They reported improved load-carrying capacity and osteoconductivity behaviors. Additional methods were proposed to enhance the mechanical properties of the printed scaffolds by manipulating the HA-TCP ratio

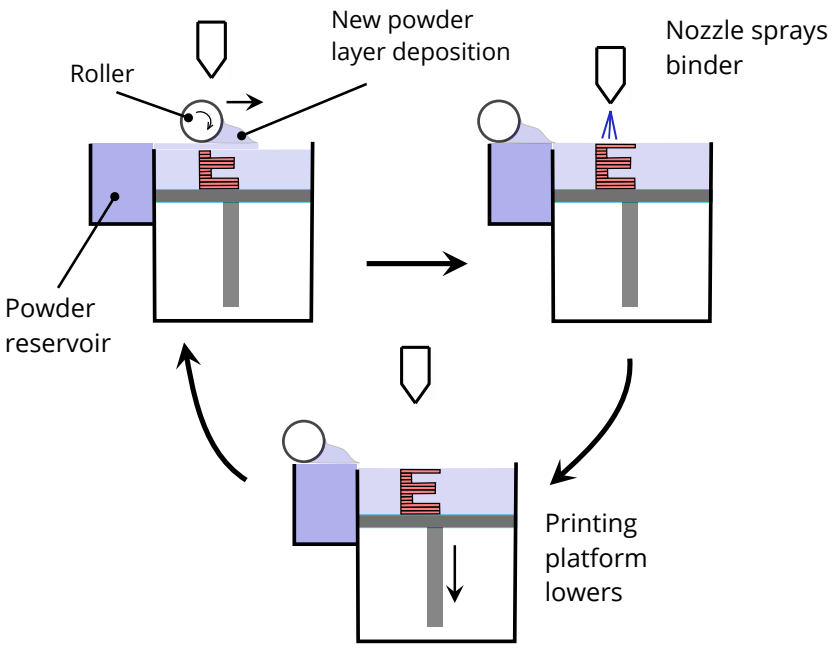


Figure 2.4: Operational principle of binder jetting printing (adapted from [24, 25])

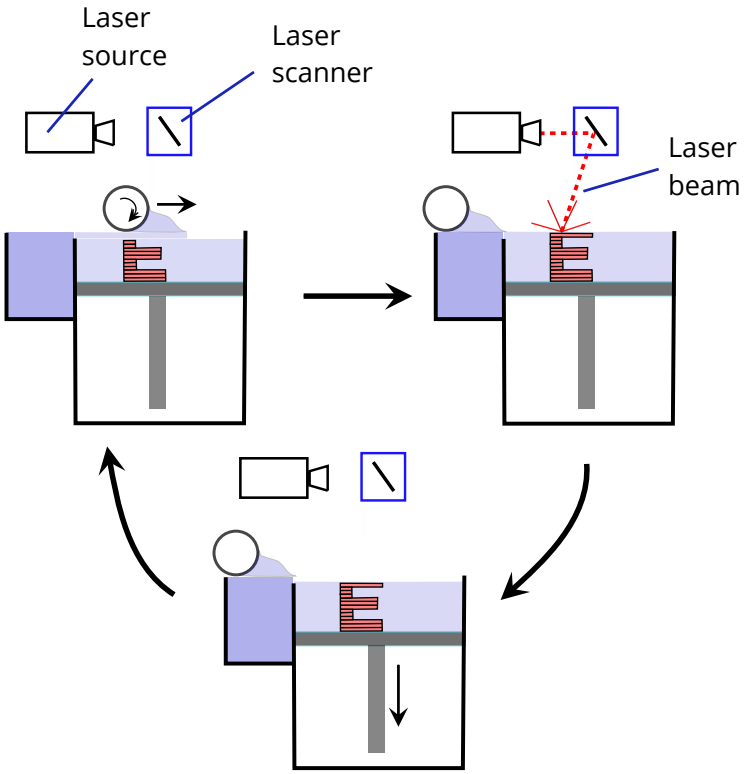


Figure 2.5: Operational principles of SLS and SLM (adapted from [25, 99, 202])

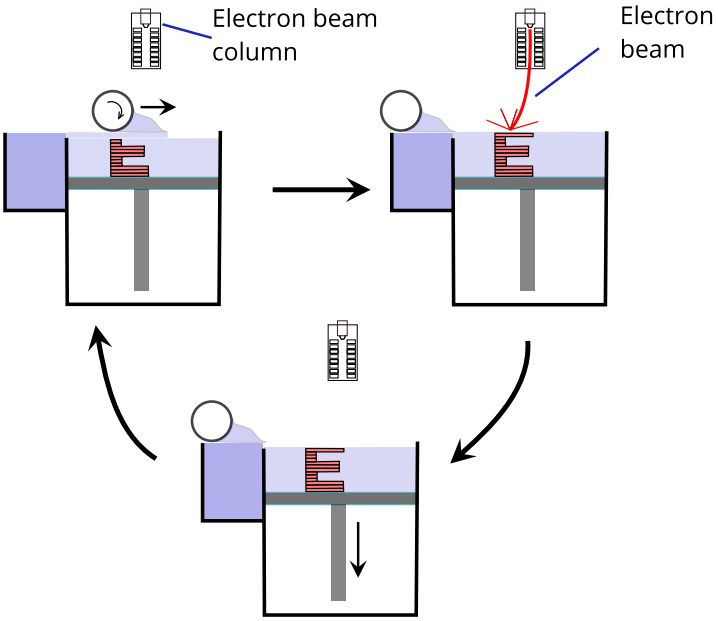


Figure 2.6: Operational principles of electron beam melting (adapted from [25])

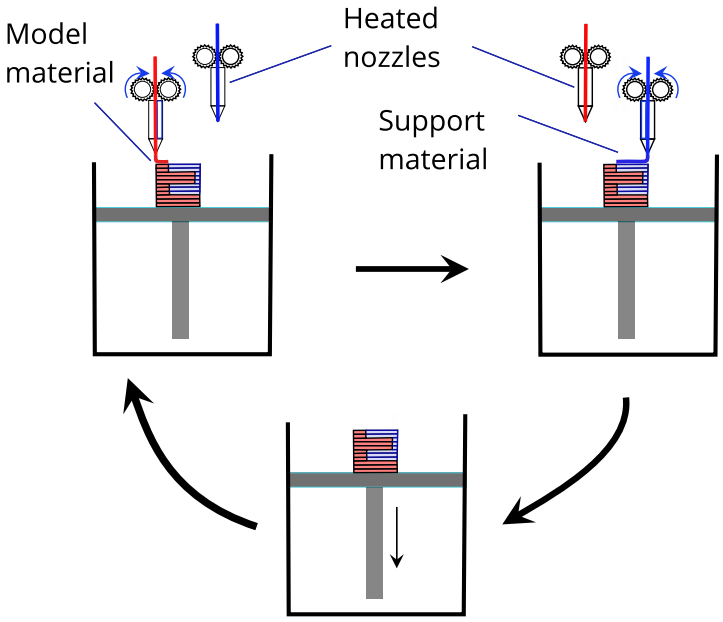


Figure 2.7: Operational principles of fused deposition modeling (FDM) (adapted from [25, 99, 203])

[214]. To accurately predict the elastic modulus of such composite scaffolds, Doyle *et al.* [215] proposed a method based on computational and experimental approaches. They first experimentally determined the elastic modulus of a construct fabricated from PCL. They later used the experimental data to determine an effective elastic modulus for the PCL/β-TCP composite scaffold. The rule of mixtures (ROM) method was employed to calculate the effective properties based on the mixing ratio and the mechanical properties of the constituent materials in the mixture. A finite element analysis (FEA) validated the calculated effective modulus. Comparing the experimental and computational results revealed that ROM overestimated the predicted elastic modulus, highlighting the need to improve this approach.

Table 2.6: Materials used in various AM processes

	Material	Pore (μm)	Process attributes	Ref.
SLS	PCL, PCL + nHA	500	Excellent biocompatibility Improved articular cartilage formation	[191, 198, 216–219]
	Polyhydroxybutyrate (PHB), Ca/P/Poly hydroxybutyrate-co-hydroxyvalerate (PHBV)	50–400	High resolution Relatively low temperature Good scaffold porosity	[85, 220]
	PLLA, carbonated HA (cHA)/PLLA	200–800	Biocompatible material Addition of HA improves osteoconductivity	[211, 221–224]
	PCL + β-TCP	300–500	Addition of 30% TCP to PCL resulted in 100% increase in compressive modulus Improved osteoconductivity when coated with collagen	[191, 214, 215]
	Polyamide (PA) + nHA	800 (40–70% porosity)	Stronger parts observed with increase of HA.	[206, 225]
	PEEK		Bioinert material with superior strength and thermal properties	[226]
	Bioactive glass additives		Improved cell activity with random porosity instead of lattice structure	[227–229]
	Nylon-12	30–50	Cell growth, differentiation, and mineralization reported. Improvement with additional chemical treatment of the scaffold	[230]
SLM	Titanium alloys: Ti-6Al-4V, Ti13Nb13Zr, Ti29Nb13Ta4.6Zr, Ti35Nb5Ta7Zr, Ti24Nb4Zr8Sn, Ti7.5Mo, Ti40Nb	250–700	Coated with PCL and PHB for improved performance (SLM) High strength, but low ductility Coated with mesoporous-bioglass (MBGs) to improve the osteoconductivity (SLM) Characterized by moderate cell seeding when compared to SLM (EBM)	[231–235]
	CoCrMo	300–900	Strong material with good biocompatibility	[236, 237]

Continued on next page ...

	Material	Pore (μm)	Process attributes	Ref.
	Zn	5–100	Biodegradable metal, with a low melting point, leading to high porosity. Improved density	[238]
	Mg	500	Good mechanical (E = 700–800 MPa), osteoconductive, and biodegradable (corrosion) properties. Extremely low cytotoxicity	[238]
	Poly(*D,L*-lactide-co-glycolide) (PDLGA) PDLGA-TCP PLLA-TCP PDLLA + β-TCP PCL-TCP PCL-HA	600–700	Complete bone regrowth was demonstrated *in vivo*	[239–241]
FDM	PCL PCL-nHA PCL-β-TCP PCL-bioactive glass (BAG) PLLA cHA/PLLA	500	High resolution Bioprinting-friendly Relatively low temperature Good scaffold porosity	[242]
	PDLLA PDLGA PDLGA- β-TCP			
	Ca/P-PHBV Ca/P cement + mesoporous silica			[238]
	Polyamide (PA)			
	Polyhydroxybutyrate (PHB)			

The osteo-performance of SLS-printed scaffolds could be enhanced by adding ceramic modifiers to the powder mix. This approach is essential since most SLS-friendly polymers are not well-known to possess these qualities. It is also possible to further enhance osteogenesis by functionalizing the scaffolds with growth factors. The growth factors are usually added later in the process through soaking to minimize the risk of thermal degradation. Saska *et al.* [85] demonstrated this by printing a PHB scaffold and later functionalizing it with osteogenic growth peptide (OGP). While improved cell adhesion and proliferation rates were obtained in response to the growth factor, no significant difference was observed in mineralization measurements, suggesting that the growth factor mainly enhanced only the initial stages of the bone modeling process. The effect was particularly pronounced on initial cell proliferation (during the first two weeks), leaving the later differentiation and mineralization stages unaffected. Metallic oxides, such as nano titanium oxide (nTiO2), are sometimes used for their antibacterial properties in SLS-printed scaffolds. A small percentage (up to 7%) addition of nTiO2 in PEEK/PGA composite scaffold, for example, can significantly reduce bacterial activity on the scaffold surfaces without interfering with cell growth in scaffolds [243]. While this is a desirable outcome to control microbial infection, the exact mechanism by which the material selectively interacts with living cells (osteoblasts and other cells included) needs to be fully understood.

The biodegradability of scaffolds used in SLS processes is another critical issue yet to be extensively probed. Ideally, the degradation rate of the scaffold must match the rate at which new tissue is generated. However, this is not always guaranteed with commonly used polymers such as PLA and PLGA. They generally degrade at different rates than the bone remodeling process. It has been suggested that metals such as [Mg] be added to produce an alkaline environment around the implants to speed up the degradation of PLLA implants. The alkaline neutralizes the acid byproducts synthesized during the degradation process, enhancing the degradation rate [243].

Selective Laser Melting (SLM)/Direct Metal Laser Sintering (DMLS). SLM employs a laser beam to melt the powder material bed (Fig. 2.5). Due to the melting involved, a more powerful laser source is required [244]. The advent of

CO_2 and fiber-based laser sources have facilitated rapid development and lower costs of SLM systems in use today [196]. Several polymers, ceramics, and metals can be processed using SLM (Table 2.6). Generally, this method is suitable for processing homogenous materials or alloys of metals (DMLS) [245]. A mismatch in material properties in a mixture could potentially result in thermal stress. Some studies, however, have demonstrated the use of SLM in the fabrication of functionally graded materials in which spatial distributions of one or more incompatible materials are printed to account for differences in the material properties [246, 247].

Since SLM involves powder melting, scaffolds made by this method have high densities, superior mechanical characteristics, and little, if any, porosity [157]. As a result, the internal microstructures are usually predefined as part of the initial scaffold design [245]. Smeets *et al.* [240] demonstrated this by designing a set of scaffolds with predefined internal geometries (pore sizes: 500–800 µm) and fabricating them from a 50:50 wt% mixture of PLA and β-TCP. They tested the scaffolds *in vitro* and *in vivo* for cell cytotoxicity and osteo-activities. No significant cytotoxicity was evident *in vitro* as a result of the material or fabrication process. The scaffolds were found to be entirely engulfed by newly developed bone cells *in vivo* with solid integration with existing adjacent bones.

SLM is also a suitable fabrication method for processing biocompatible metals [248]. For example, the low boiling point of Zn allows the formation of bubbles inside the construct during printing, contributing to the formation of porosity inside the scaffold [249–251]. Magnesium alloys have often been fabricated and investigated for bone scaffolding due to their excellent biocompatibility and strength [248, 252]. Titanium (Ti) and titanium alloys such as Ti6Al4V are also widely used in bone-related implant applications [209].

Although the high processing temperatures in SLM are the reason for the superior mechanical properties of its constructs, they prohibit the addition of biological growth factors and drugs into the material mix [25]. This limitation is at odds with the ever-growing interest in incorporating drugs and growth factors into the scaffold material. In addition to the de-powdering issues common to powder bed printing processes,

constructs manufactured by SLM also require heat treatment to remove the thermal stresses caused by repeated heating and cooling cycles.

Electron Beam Melting (EBM). *Electron Beam Melting* is very similar in its working principle to SLM. It is mainly used to process the same category of materials as SLM (Table 2.6). The main difference between the two is that, instead of a laser, EBM uses an electron beam as a thermal energy source to melt the powder material (Fig. 2.6). It has better resolution and accuracy than laser-based fabrication systems [99]. However, only electrically conductive materials (metals) can be processed by EBM [47]. Like SLM, EMB can fabricate constructs with superior mechanical strength [203]. Several groups have used it to study cell attachment and subsequent proliferation on implants made from Ti alloys (Ti-6Al-4V) [234, 253–256]. Although titanium-based scaffolds are non-degradable, they are suitable for cell seeding and cultivating a living tissue *ex vivo* or implanted at specific sites to allow *in situ* bone tissue deposition. Like SLM, the melting in EBM has a detrimental effect on process-induced porosity, mainly resulting in solid internal microstructures. Therefore, required scaffold porosities are designed or acquired using digitizing equipment [235, 257].

Material Extrusion

In material extrusion, a single or a mixture of materials in viscous or amorphous consistency is ejected out of a nozzle tip under pressure. The extruded material is deposited layer by layer in a controlled manner to create 3D constructs (Fig. 2.7). Unlike powder-based systems, this process does rely on unactivated material to support overhanging features in the construct during the printing process. For this reason, printers using this process, except for wet-spinning-based systems, usually employ a secondary nozzle to deposit support material wherever needed. Compared to the primary model material, the support material is designed to have low mechanical strength, so it can be easily broken off the model and removed once it serves its purpose. Alternatively, some support materials are designed to be dissolved using chemical solvents that target only the chemical composition of the support material. Material extrusion may be broadly categorized into three groups based on the starting

state of the material or the method employed to bring the material into a liquid or semi-liquid consistency. They include: Methods relying on thermal energy sources to melt the material before extrusion (*Fused Filament Fabrication*); Methods based on wet-spinning wherein a liquid material is injected into a liquid bath for coagulation (*Wet-spinning Based Extrusion*); and Processes involve liquid-based suspension and thinning of the base material by using water or suitable solvents before extrusion (*Direct Writing/Robocasting*). Table 2.6 lists materials used by these processes to fabricate bone scaffolds.

Fused Filament Fabrication (FFF). In FFF, the material comes directly out of a nozzle tip partially melted. The thermal bond created between the successive layers is due to the fusion at the contact points of the deposited beads, hence is not as strong as a fully melted connection. As a result, constructs produced by this method display anisotropic mechanical behaviors [258]. FFF is best suited for processing polymers because of the thermal melting involved. Several studies have used this fabrication method with biodegradable polymers such as PCL, PLA, PLLA, and PDLGA for bone scaffold applications with encouraging results [53, 186, 259–262]. However, this method cannot handle pure osteoinductive/osteoconductive ceramics such as HA and TCP. As a workaround, ceramic and bioglass minerals are usually added in a small percentage to the polymer base [163, 172, 239, 263–265]. This practice has improved the differentiation and mineralization properties of polymeric bone scaffolds. The other major drawback of this method is the detrimental effect of high temperatures on bioactive additives. Because of this limitation, FFF scaffolds are usually coated with bioactive agents post-fabrication.

Wet-spinning Based Extrusion. In wet spinning, a liquid polymer stream is injected into a coagulating solvent bath at a controlled rate to create a solidified fiber inside the bath. Since wet spinning is a low-temperature application, it addresses most of the temperature-related challenges associated with fused filament fabrication. Like the other fabrication processes, polymers used in wet spinning can also be mixed with other ceramic and bioglass fillers to enhance the biological and mechanical properties of the resulting scaffolds [266–268]. The incorporation of temperature-

sensitive bioactive agents directly into the material mix has been demonstrated using non-computer-aided wet-spinning fabrication methods with promising results [269, 270]. For example, Rodrigues *et al.* [271] explored this idea by wet-spinning a corn starch/PCL scaffold functionalized with Si-OH groups. The PCL was first obtained by dissolving in chloroform, after which a PCL and corn starch solution was injected in a methanol bath in a controlled manner to produce fiber strings. *In-vitro* evaluation has indicated that the sample scaffold could support differentiation of MSC cells, followed by mineralization within 14 days.

Recently, an effort has been underway to adapt this fabrication method into AM by mounting the extruder on a multi-axis-controlled printing head. The resulting configuration is commonly referred to as computer-aided wet spinning (CAWS). Puppi *et al.* have conducted several investigative studies in this area [272–274]. They employed this method to fabricate bone scaffolds from commonly used synthetic and natural biodegrade polymers such as PCL and PHA [266, 274]. They reported improved control and accuracy over fiber alignment and internal architecture, including internal porosity of 79–88%, fiber diameter 47–76 µm, and pore size 123–789 µm. *In vitro* measurements of cell proliferation and differentiation measured on a PHBHHx scaffold using MC3T3-E1 preosteoclast cell line have shown promising results after 21 days of culture. Another *in-vivo* study by Dini *et al.* . [272] has shown encouraging results in cytotoxicity and osteogenetic activities. The scaffold implants, made from pure PCL and PCL/HA blend for comparison, exhibited no toxicity and were found to support bone regeneration 24 weeks after implant, with a slight edge in performance in favor of HA presence.

Robocasting. In the context of bone TE, robocasting involves extruding and depositing layer by layer a slurry of the material, usually prepared by suspending ceramic powders in water-based solvents, to create 3D scaffolds. The content of the slurry is carefully selected to achieve targetted mechanical and biological behaviors from the resulting scaffold.. The thinning medium serves the double purpose of transforming the powder into a slurry consistency while binding the powder particles together during printing. While water is widely used with HA and TCP, other bioglasses,

such as 45S5 Bioglass, require specialized additives such as carboxymethyl cellulose (CMC) [275]. Water is ideal because, unlike phosphoric acid and other chemical-based solvents, it leaves no residual toxicity and is compatible with bioactive agents. Satisfactory mechanical results for water-based robocasted scaffolds have been reported for HA and TCP scaffolds [276–278]. The low temperatures in the process alleviate the concern of thermal degradation of bioactive additives.

Material extrusion is by far the most popular fabrication method used in TE, and it is frequently used in combination with other methods to improve the attributes of fabricated scaffolds. Dalton *et al.* [279] and Mozetic *et al.* [280] have employed FFF and electrospinning to fabricate multimodal constructs. Interweaving the electrospun fibers between the FFF layers provided unique internal architecture favorable for biological activities. The constructs combined the benefits of both processes. While extrusion from the FFF provided good overall strength and mechanical properties, the intermediate electrospun microfibers enhanced cell adhesions and subsequent proliferation performance of the scaffold.

Vat Photopolymerization

Vat photopolymerization is an additive manufacturing process in which photosensitive materials (photopolymers) in liquid or viscous states are selectively cured using light. The method is used in bone scaffold fabrication by mixing typical non-photosensitive scaffold materials with known photopolymers [281]. Table 2.7 is a list of common materials used in this method. Depending on the type of light and mode of exposure, this process can be further categorized into stereolithography (SLA), digital light processing (DLP), and continuous digital light processing (CDLP).

In SLA (Fig. 2.8), a vat of photopolymer liquid is exposed to a focused laser beam at the surface level. Material scanned by the laser goes through the photopolymerization process and eventually hardens to become part of the solid construct [282]. Printing usually starts by positioning the support plate (print bed) under the liquid surface. The first layer is directly printed onto this plate. After exposing the first layer, this plate is lowered by a distance equal to the thickness of the next layer. Material not cured by the laser remains in liquid form. Unlike powder-based systems, unexposed

material in the vat cannot be used as support material. Because of this, additional structures are printed as part of the printing process to support overhanging features [37].

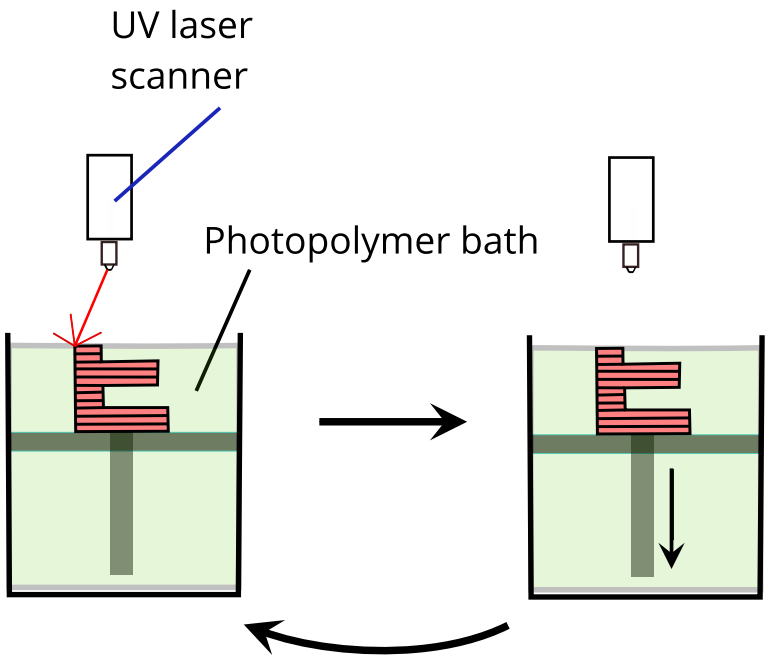


Figure 2.8: Stereolithography (SLA) (adapted from [25, 99, 203])

The print resolution in SLA depends on, among other factors, the laser diameter and layer thickness (table step) used in the process [6]. Generally, layer thickness in the 50–250 µm range have been reported [6, 37]. The absence of high temperatures, common to powder-based AM, is credited for the smoother surface finishes in SLA constructs. Despite SLA's high resolution and accuracy, its use in bone TE has yet to be fully realized. This is mainly due to the limited availability of biocompatible photopolymers [283]. Mixing well-established scaffold materials, HA in particular, with photopolymers has yielded encouraging results [284]. However, this approach has its limitations. A study by Scalera *et al.* [285] has indicated that increasing the percentage of HA suspension in the slurry mix (from 0 to 60%) could increase the viscosity of the slurry and slow down and the speed of the photopolymerization reaction. Additional heat treatment and sintering may be required to achieve improved mechanical

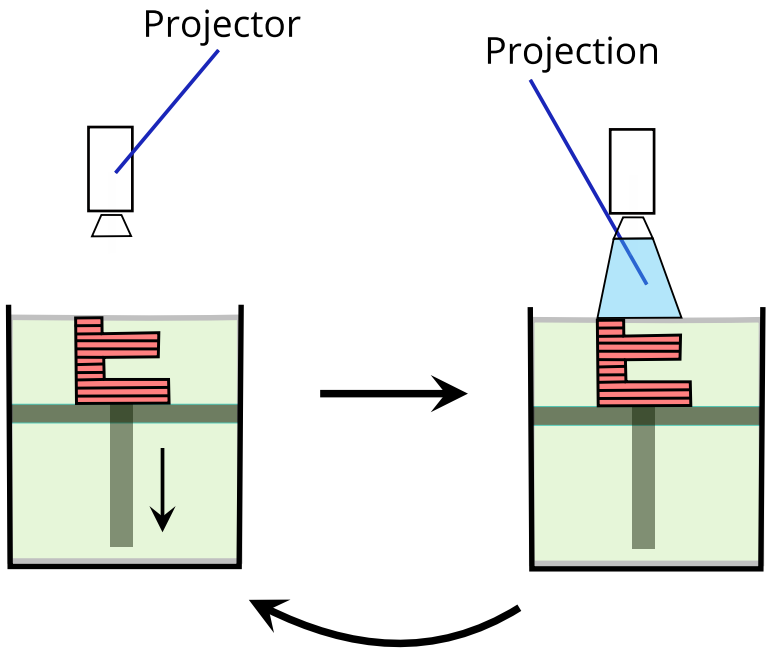


Figure 2.9: Digital light processing (DLP). (adapted from [25, 99])

properties. Table 2.7 summarizes common scaffold materials and photopolymers used in SLA.

Instead of a laser, digital light processing (DLP, Fig. 2.9) uses UV (*blue light*) projections of the object's cross-sectional images for curing the liquid photopolymer [282, 286]. Compared to SLA systems, DLP has better resolution (∼70 µm) and significantly faster processing speeds [99]. The UV-unresponsive HA and TCP could be mixed with other biocompatible photopolymers to improve their processability [287]. Using DLP, Tesavibul *et al.* [288] have printed bone scaffolds from a mixture of HA and a photosensitive resin (methacrylate-based monomers 98%, photoinitiator 2%).

Continuous Digital Light Processing (cDLP) takes DLP one step further by using continuous digital projection of successive layer images to create a smoother transition from one layer to the next. For all practical purposes, this process can be assumed to have no layers at all. As shown in Fig. 2.10, the continuous projection and the

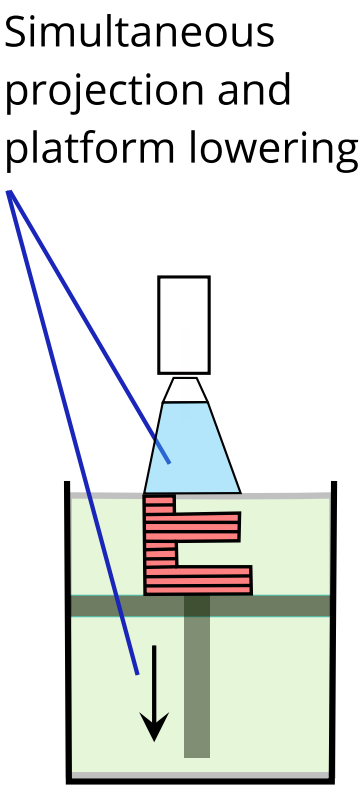


Figure 2.10: Continuous digital light processing (CDLP). (adapted from [99])

printing table motion happen synchronously. The result is a construct with high resolution (< 50 µm) and smoother surface finishes [289]. Digital steps observed in SLA and DLP are absent here. This method uses similar photopolymers as SLA and DLP.

In SLA, DLP, and cDLP, photopolymerization takes place in the presence of light and photoinitiators. Solvents and dye-initiators are sometimes needed to reduce the viscosity of the photopolymers and calibrate the depth of polymerization, respectively [289]. A discussion on the effects of solvents and dye-initiator concentrations on the depth of curing is presented by Dean *et al.* [290]. Table 2.7 is a list of commonly used photoinitiators and photopolymers.

Some fundamental challenges impede the full realization of vat photopolymerization for clinical applications. Most of these challenges stem from a shortage of photochemicals that meet clinical requirements. Most photopolymers, photoinitiators, dye-initiators, and solvents pose serious cytotoxicity risks. In addition, only a handful of materials satisfy the mechanical strength requirements [291]. Poly(propylene fumarate) (PPF) has evolved as a leading material of choice for cDLP fabrication of bone scaffolds [289, 292].

Table 2.7: Biodegradable photopolymers and photoinitiators used in SLA (adapted from [285, 291, 293, 294]

Biodegradable photopolymer	**Photoinitiator (trade name)**	**Wavelength peak**
	Irgacure 2959	257–276 nm
	Irgacure 184	246–3 33 nm
	Irgacure 651 DMPA	250, 340 nm
Methacrylated gelatin	Irgacure 369	233, 324 nm
Polypropylene fumarate (PPF)	Bisacylphosphine oxide (BAPO)	—
Linear poly(D,L-lactide)-methacrylate	Irgacure 819	—
Poly(TMC-co-CL)-coumarin	Camphorquinone	285, 400, 500 nm
Star-shaped poly(D,L-lactide)-fumarate	LAP	375 nm
	Lucirin TPO-L	380 nm
	VA-086	385 nm
	Eosin Y	514 nm
	BAPO	350–430 nm

2.5.4 Electrospinning

Electrospinning is a general term used to describe a process in which a steady stream of an electrically charged polymer in a solution or viscous state is drawn into a fiber under the actions of electrostatic forces [11].

In a typical configuration, the material source (spinneret) and the fiber collector platform are connected to the opposite terminals of a high-voltage source. The potential difference between the spinneret and the collector, usually in the 5–30 kV range, generates an electric field in the air gap between them. The shorter this gap, the stronger the electric field. The spinneret tip is carefully located inside this electric field, where the material is injected in a controlled manner.

The liquid polymer is usually pumped out of the spinneret at a relatively low flow rate (2–5 $\mathrm{mL\,h^{-1}}$ range)(Fig. 2.11). A droplet-like accumulation first occurs at the tip of the spinneret due to the low flow rates involved. The droplet becomes electrically charged when the high-voltage supply is turned on and behaves like a charged particle within an electric field.

Because they have similar polarities, the electrical charges inside the droplet push each other apart, resulting in dense charge concentration on the outer surface of the droplet. Because the electric field pulls the charge-carrying material toward the collector plate, the surface charge concentration is notably higher in droplet regions facing the collector platform. Increasing the applied voltage leads to the deformation of the droplet due to interactions among the Coulomb forces, the electric field forces, and the surface tension forces that try to hold the droplet together. Further voltage increase eventually changes the droplet's shape to a cone, with its circular base matching the diameter of the spinneret tip and the apex pointing towards the collector platform. This phenomenon is known as Taylor cone formation, named after Sir Geoffrey Taylor (1886–1975) as a tribute to his significant contribution to research in this field.

Until a critical voltage is reached, the surface tension forces effectively counteract the electric field and Colomb forces that try to break the droplet's surface tension. Increasing the applied voltage beyond this critical value breaks the surface tension and initiates a polymer jet. As long as the high-voltage supply is maintained, the jet flows continuously to become the electrospun fiber. This process can fabricate fibers with diameters in the nano/micro-scales. This process could provide exquisite control over some critical features in scaffold fabrication.

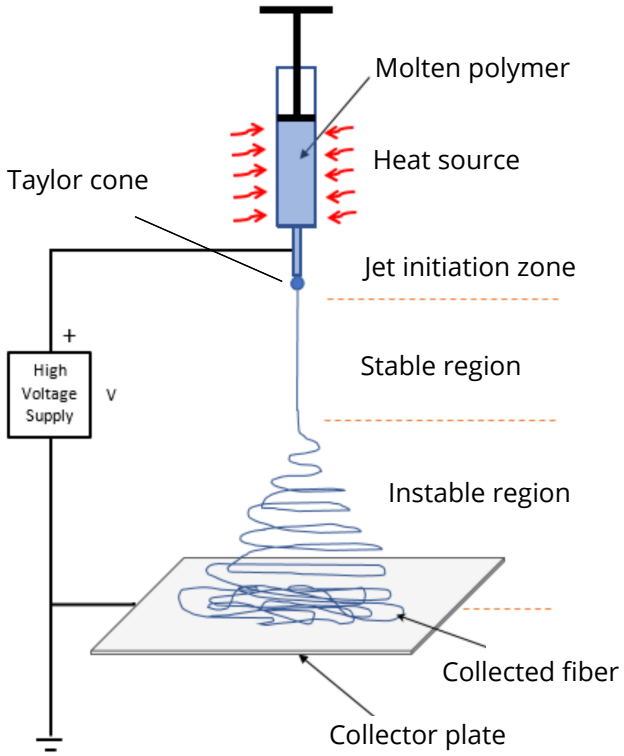


Figure 2.11: Schematic diagram of MES

Solution-electrospinning. The use of solvents in solution-electrospinning (SES) is known to reduce surface tension in the polymer material. This reduction in surface tension has two implications. First, the material easily flows and electrospins because of the low viscosity associated with the solvents. As a result, solution electrospinning is well known for producing fibers with diameters in the nano-scale (Fig. 2.12) [295].

Second, the jet flow is characterized by a lateral "whipping" motion. Residual solvents remain in the fiber, carrying electrical charges on the fiber body. This repulsive Colomb forces associated with the extra charges and the lower viscosity impart wiggling motion to the fiber (*whipping motion*). Several methods have been proposed to reduce the chaotic nature of fiber deposition. A shorter fiber travel gap between the spinneret and the collector has been found to positively affect the whipping amplitude. However, because of length limitation, the fiber would not go through the complete stages of drawing to the final diameter. Other methods have used auxiliary electrodes to apply lateral electrostatic forces to dampen the fiber's oscillation in the

lateral direction [296–298].

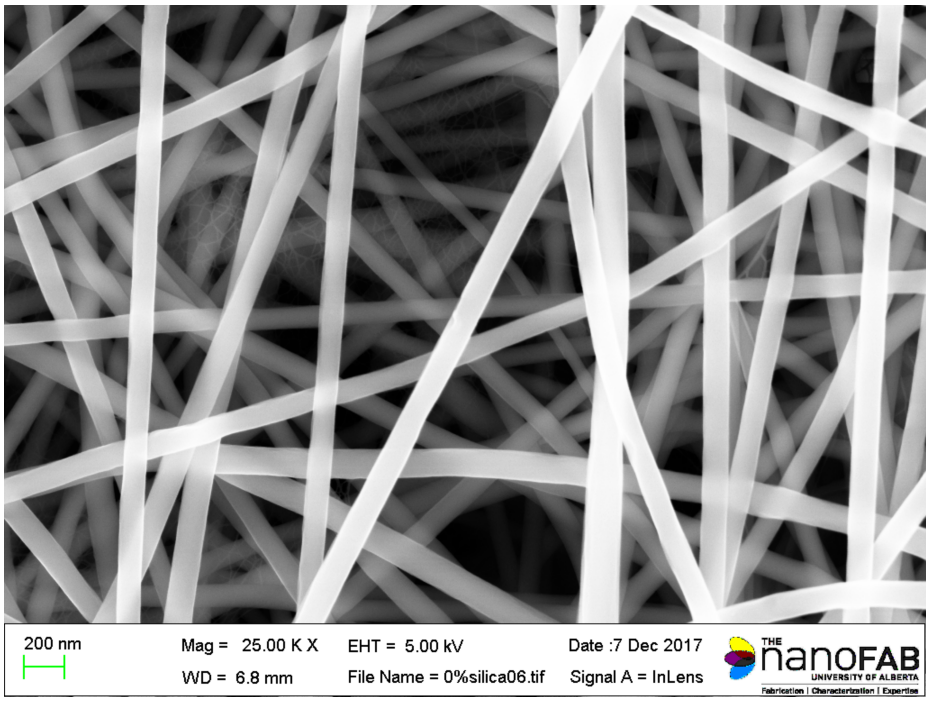


Figure 2.12: Electrospun Nylon 6 nanofiber

Solvents used in solution electrospinning are usually designed to evaporate when the material passes through the air gap between the spinneret and collector plate. However, complete evaporation is not always guaranteed, sometimes leading to the accumulation of residual solvents on the collected fiber mesh [299]. Residual charges associated with the residual solvent on the collected mesh create repulsive forces between successive layers that weaken inter-fiber bonds. As a result, this process is limited to producing softer fiber mats up to 4 mm thick. The other concern with residual solvents is their potential detrimental toxic effect on cells and tissues when used for TE [279].

Despite these technical limitations, solution electrospinning has been extensively studied for TE applications [300, 301]. The extremely small fiber diameters and the tight deposition arrangements produced by SES are attractive. These qualities have made the process a potential candidate for addressing some limitations other scaffold fabrication processes face.

One of the main requirements outlined for the success of bone scaffolds is the

size and geometry of porosity. Venugopal *et al.* [301] have used SES to produce highly porous scaffolds from a bio-composite mixture of PCL, nHA, and collagen (PCL/nHA/col). The scaffolds were characterized by pore sizes of 2–35 µm and porosity levels of more than 80%.

The other great advantage is that the random fiber depositions in this method produce surface roughness suitable for cell attachment in bone scaffolds. Several studies have already demonstrated the suitability of electrospun mats for initial cell adhesion [302]. Table 2.8 is a list of materials investigated for such applications.

Table 2.8: Common bone scaffold materials used with electrospinning

Material	**Fiber diameter**	**Process attributes**	**Ref.**
PCL PLA PGA PLLA/PEG PCL/gelatin Bioactive glass/PLLA PCL/nHA/Collagen	100 nm–6 µm	Relatively poorer cell adhesion in the pure form of polymers Improves with the addition of HA Highly porous Relatively slower production rate Extremely small fiber diameter	[301, 303]

Melt-electrospinning. When a thermal source, instead of a solvent, is used during electrospinning to change the state of polymer to a viscous fluid, the resulting process is called melt-electrospinning (MES) [12–14]. Because of the absence of solvents, the process does not exhibit a significant reduction in surface tension, a common problem in SES [304, 305]. Larger fiber diameters and wider inter-fiber spacing characterize the depositions produced by this method. Melt-electrospun mats perform inadequately with initial cell adhesion. However, unlike SES, they excel at inward cell growth and display good mineralization gradients [306]. The lateral whipping motion is significantly reduced in melt electrospinning due to minimized residual charges on the fiber [307].

Additionally, melt electrospinning is characterized by improved inter-layer bonding because residual solvents and associated residual electrical charges are absent. Combined with faster spinneret translational velocity, these characteristics allow a more orderly deposition of fibers on collector plates (Fig. 2.13). These improvements permit a setup in which the spinneret-collector distance can be increased significantly.

There has been a growing interest in developing a melt electrospinning-based AM

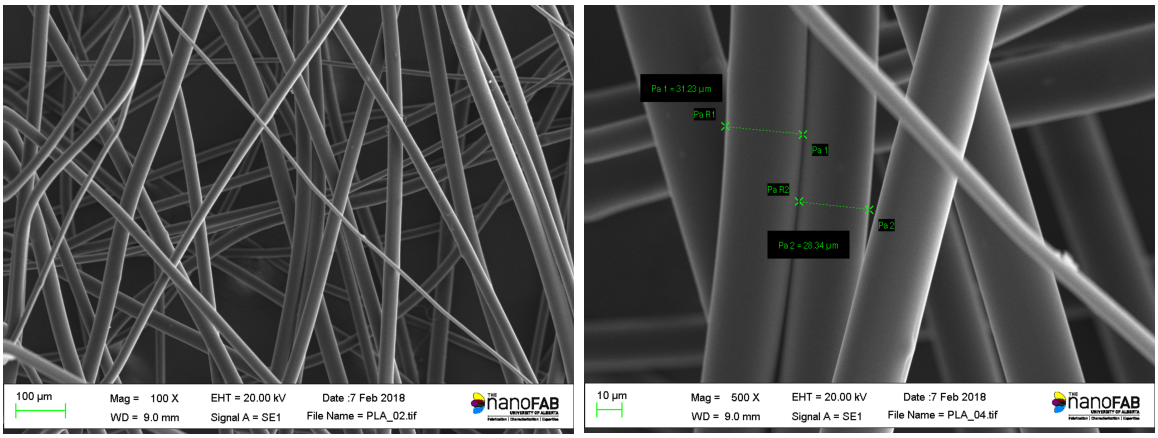


Figure 2.13: Melt electrospun PLA fibers

process capable of building accurate 3D structures [308]. Achieving this goal, however, requires addressing some fundamental challenges and understanding the interactions among the process parameters, such as applied voltage, collector-spinneret gap, melting temperature, and material properties [296]. Both experimental and theoretical modeling efforts have been pursued to this end [309–311]. For example, Bentolhoda *et al.* [312] used the response surface modeling (RSM) method to primarily predict and optimize the porosity of nanofibers collected in a solution-based electrospinning setup. The change in porosity (a response variable) was observed for independent process parameters (factors) changes. The resulting relationship between the porosity and the independent variables was later fitted into a regression model:

$$Y = 39.94 - 1.11X_1 + 3.24X_2 + 2.46X_3 - 1.38X_4 + 1.15X_1X_2 - 1.62X_1X_4 + 1.38X_2X_4 + 1.19X_3X_4 + 3.10{X_4}^2, \quad (2.4)$$

where Y, X_1,X_2, X_3, and X_4 are the porosity of the fiber mat, solution concentration, applied voltage, tip-to-collector distance, and volume flowrate, respectively.

The effect of increasing the spinneret-collector gap on overall spinning performance also needs to be fully understood to adapt electrospinning successfully. For example, to establish a groundwork in this area, Joo *et al.* [306, 313] have modeled the MES as a non-isothermal polymer jet based on the equations of material continuity, conservation of momentum, charge, and electric field.

$$\text{Continuity: } \pi R^2 v = Q \tag{2.5}$$

$$\text{Momentum: } \rho v v' = \rho g + \frac{F_T}{\pi R^2} + \frac{\gamma R'}{R^2} + \frac{\sigma \sigma'}{\varepsilon_o} + (\varepsilon - \varepsilon_o) E E' + \frac{2\sigma E}{R} \tag{2.6}$$

$$\text{Charge: } \pi 2 R^2 K E = 2\pi R v \sigma = I \tag{2.7}$$

$$\text{Electric field: } E(z) = E_\infty(z) - \left[\frac{1}{\varepsilon_o}(\sigma R)' - \left(\frac{\varepsilon}{\varepsilon_o} - 1\right)\frac{(ER^2)''}{2}\right] \ln\left(\frac{d}{R_0}\right), \tag{2.8}$$

where R is the jet radius, v is average fluid velocity, Q is material flow rate, ρ is fluid density, g is acceleration due to gravity, F_T is tensile force, γ is surface tension, σ is surface charge density, e is dielectric constants of the plastic, ε_o is dielectric density of the air, E is the electric field, K is electrical conductivity, I is the current, E_∞ is applied electrical field, d is collector-spinneret distance, and R_0 is spinneret diameter.

Significant effort is underway to understand and accurately predict the whipping instability zone [314]. Reneker *et al.* [315] developed a theoretical model for the MES process based on a one-dimensional viscoelastic flow. They used this model to represent the viscoelastic forces between the discretized fiber segments of mass m and electrical charge e. In addition to the viscoelastic forces, they also considered Coulomb repulsion forces, the electric field forces, the surface tension forces, and the perturbation triggering the instabilities. The equation of motion was formulated for each bead using Newton's second law:

$$m\frac{d^2 \boldsymbol{r}_i}{dt^2} = \sum \boldsymbol{F}_i = \boldsymbol{F}_{Ci} + \boldsymbol{F}_{Ei} + \boldsymbol{F}_{vei} + \boldsymbol{F}_{capi} + \boldsymbol{F}_{gi}, \tag{2.9}$$

where $\boldsymbol{F}_{Ci}$, $\boldsymbol{F}_{Ei}$, $\boldsymbol{F}_{vei}$, $\boldsymbol{F}_{capi}$, and $\boldsymbol{F}_{gi}$ in Eq. (2.9) are, respectively, the Coulomb, electric field, viscoelastic, capillary, and gravitational forces.

Table 2.9: Common materials used with melt electrospinning

Material	**Fiber diameter** (µm)/ **Pore size** (µm)	**Process attributes**	**Ref.**
PLA, PLA-PEG-PLA, PLA + 45S5 bioglass	Diameter: 31	10 kV, 5 mm collector distance	[316]
PLLA, PLL-PCL	Pore :220–260		[317]
PCL, PCL - Strontium-substituted bio-active glass (SrBG)	Pore: 250–300	11–12 kV, 200 mm collector distance	[18, 318, 319]
PLLA	Diameter: 10	40 kV, 18 cm	[320]

Direct Writing with Electrospinning As the trend in TE accelerates towards the fabrication of scaffolds with customized and precisely controlled internal and external architectures, much interest has been generated to integrate the spinneret and

collectors with multi-axis-controlled platforms. *Direct writing* has become a term commonly used to describe an electrospinning process in which the movement of either the spinneret or the collector plate is controlled by a computer [280, 321–323]. Both SES and MES processes could benefit from this setup to improve the scaffold's cell adhesion and inward infiltration properties. The advantage of this approach is that the fiber mat collected will have both good cell adhesion and inward growth and mineralization capabilities. More than one spinneret could also be used in the setup to create what is known as multimodal electrospinning [324]. Because of the extreme whipping and deposition instabilities observed in solution-based electrospinning, faster translation speeds are required to get straight fiber depositions on the collector plate. On the other hand, the MES, because of its higher viscosity, does not suffer from this limitation and allows straighter depositions at relatively lower translational speeds [279].

While not a desirable property for other applications, the whipping effect could play a constructive role in bone TE as it helps create the randomness in fiber depositions, resulting in inter-fiber spacing that closely resembles the bone ECM.

2.6 Translating Scaffolds for Clinical Use

The adaptation of various fabrication approaches to clinical use remains a significant challenge. Despite many pre-clinical studies on bone regeneration, only a few truly-fabricated scaffolds have found their way into clinical utility. Digital technologies like beam computed tomography or MRI allow clinicians to analyze the shape and size of bone defects and then precisely plan and fabricate 3D scaffolds before the surgery. The ability to generate 3D bone grafts by ink and the filament-based multi-material 3D printer has opened up various possibilities. Other computer-aided manufacturing (CAM) methods are also employed in this effort. Luongo *et al.* [325] synthesized micro/macroporous BCP scaffolds consisting of 70% TCP and 30% HA using a proprietary CNC milling machine for the treatment of partial and total edentulism. Defects no larger than 12 mm in height and 10 mm in width were treated. The results showed excellent fitting of the scaffolds and new bone formation after eight months of implantation. Yang *et al.* [326] synthesized patient-specific surgical

plates by 3D printing to treat resected head and neck tumors. The plates were printed with Grade 2 pure Ti using the SLM fabrication method and used to treat bone ten patients. After the treatment, the bony and soft tissue margins tested negative for tumor cells in all ten cases, while a few cases were referred to post-radiotherapy or chemotherapy treatment. No significant complications were recorded 6.5 months after the surgery. These outcomes are promising, but the narrow range of biomaterials available for bioprinting is bound to limit clinical testing in the long term. New printable materials with good biocompatibility, mechanical properties, and the ability to deliver bioactive factors could support bone regeneration.

2.7 Perspective and Future Directions

Not long ago, bone repairs were done using only bone tissues harvested from the patient's body or compatible donors. Because of issues related to the shortage of tissue supply, donor site morbidity, and incompatibility, these treatment options were mostly limited to small-size defects. There has been an ever-growing interest in 'solutions' addressing these limitations. Significant progress has been made over the last two decades. New fabrication methods and biocompatible materials are making a significant impact. Initial efforts were mainly focused on mimicking the macrostructure of bones for permanent implant purposes. Recent efforts, however, are mobilized toward fabricating scaffolds with suitable materials, internal microstructures, and bioactive factors to create ideal conditions for tissue regeneration.

Four general scaffold requirements have been identified. The first is a geometrical requirement. Bone scaffolds must have optimally sized and interconnected porous spaces to house appropriate cells and allow the transportation of nutrients and metabolic wastes to and from the cells. A suitable pore architecture gradient is also at the early stages of exploration. While the outside contours of the defect site are a good design criterion to follow, it needs to be clarified if mimicking the original pore structure of the bone is the right approach. Regenerative events may not be optimal with the architecture of the mature tissue. The second requirement is related to the properties of the material from which the scaffold is made. These materials must be osteoconductive, biocompatible, and biodegradable. The material

is vital in the initial cell attachment and later cellular penetration and mineralization stages. The third requirement relates to the mechanical properties of the scaffold. The scaffold needs to meet the load-carrying requirements at the implant site. While mechanical strength is a critical requirement in some implant sites, others are less demanding, and healing has been achieved using particulate implants lacking a cohesive macroscopic mechanical strength. Finally, scaffolds are expected to bear bio-active molecules capable of modulating cellular activity in and near the implant site.

The AM binder jetting and material extrusion systems are characterized by relatively slower processing speeds and low print resolutions due to the inherent nature of the physical motion of their nozzle heads. The binder-wetting mechanics also limits the minimum resolution achievable, below which it is practically impossible to draw a clear boundary between the wet and dry regions. Unfortunately, because the minimum achievable resolution from these processes is too large for some critical ECM features, these two fabrication methods are not ideal for mimicking bone ECM. This assertion is partly based on the observation that most of the literature reviewed has attempted to use these two systems to fabricate constructs with predefined internal geometries. The major drawback was that most of the designed internal architectures were based on primitive geometries, such as square, circular, or some regular polygon-shaped voids patterned throughout the volume of the scaffolds. In contrast, the actual bone ECM micro-structures are much lower in resolution and complex in shape (often with graded features) than what most of the studies reviewed here were able to demonstrate.

Binder jetting and fused deposition modeling can be combined with other fabrication methods to produce the initial stage scaffolds in multi-hierarchical configurations. Gas foaming of scaffolds initially printed using FDM systems is a good example. Gas foaming also involves relatively low temperatures, allowing the incorporation of sensitive factors into the materials mix. Table 2.10 presents a summary of these manufacturing processes and materials.

Although powder bed fusion systems have better definition and strength than binder jetting and material extrusion systems, the high level of temperatures involved

in their process, except for SLS, renders the impregnation of sensitive growth factors almost impossible. This limitation is at odds with the current trend of incorporating such additives into the material substrate. These systems also suffer from powder trapping if the internal porosities were to be created from a pre-designed or digitally scanned geometry. Most of the reviewed literature needed additional information in this area. Since simplified internal geometries and coarse resolutions were adapted in these investigations, the issue of powder trapping has yet to come up as a real challenge. The real power of AM fabrication systems such as SLS, 3DPP, and PBF lies in the precise selection of powder particle sizes. The pore connectivity and internal micro-architecture must be a byproduct of the interparticle spacing. It should come from something other than the 3D-modeled void space achieved by depowering.

Based on our review, vat polymerization-based fabrication methods have the best outcome regarding printing resolution and processing speed. However, the limited availability of biocompatible photopolymers poses a significant challenge. Adding ceramic fillers to get desirable material properties has been reported to improve strength. However, it is also observed to have a detrimental effect on the photopolymerization process, limiting the maximum amount of ceramic additives that can be safely mixed with the photopolymer.

The random whipping motion in electrospinning processes, integrated with a multi-axis motion control system, could produce 3D scaffolds with spatially-varied internal microstructures. Achieving this, however, requires addressing the thickness limitations. The process in its current state produces mats only in the 5–10 mm thickness range [327]. The complex interactions among its process parameters make controlling and maintaining constant fiber diameter challenging. For this reason, until recently, the process was only exclusively used to fabricate textile-like structures. Due to the interactions between electrostatic forces and viscoelastic forces, the process is characterized by random fiber deposition patterns on the collector plate. The amplitude of lateral whipping increases as the fiber accelerates toward the collector plate. The resulting collection diameters measured in standard SES and MES setups are too wide to produce a reasonable scaffold definition on the macro scale. The whipping amplitude must be in a specific range to satisfy the following two requirements: 1) the

magnitude of the whipping amplitude must be large enough to create a localized random fiber orientation when observed at the micro level, and 2) the amplitude should be small enough to create a reasonably well-defined 3D scaffold shape on the macro level. With increased print thickness in the z-axis direction and dampened whipping instability, MES could fabricate bone scaffolds out of fiber depositions. This effort requires a thorough understanding of the behaviors of the process parameters and the effects of their interactions on fiber morphology.

Table 2.10: Summary and comparison of fabrication methods and materials

Process		Resolution	Materials	Properties	Ref.
Gas foaming		40–800 µm	Biopolymers: chitosan, alginate, gelatin Synthetic polymers: PCL, PLA, PLLA, PLGA Polymer/Ceramic composites: PLA/HA, chitosan/HA, PLGA/HA	High porosity scaffold Internal microstructure not explicitly defined Well-established system Could be integrated with other AM methods to increase microporosity Relatively faster than most AM processes	[42, 115–117, 119, 132, 328]
Cryotropic gelation		20–400 µm	Polymers: Chitosan, collagen, gelatin, PVA, PAA, PEG, PHEMA, PTCA Composites: Gelatin/HA, collagen/HA	Crosslinking required for resilient scaffolds, hence limited material availability Open to blending to a certain extent Well-established method	[123, 130, 144, 329, 330]
Additive Manufacturing	Binder jetting (3DP)	10–30 µm	Ceramics: HA, α-TCP, β-TCP, bioglass Composites: HA/TCP, HA /Polymer, TCP/PCL or PLA, PEG, PLGA, gelatin, chitosan, and collagen.	Widely applicable method Low to high-temperature application Suitable for processing bioceramics Binder (water/chemical) involved; hence risk of toxicity Slow build speed	[4, 41, 54, 64, 69, 330, 331]
	Selective laser sintering	50–100 µm	Polymers: PCL, PEEK, PLLA, PGA, PLDLA, PVA Composites: PCL/TCP, PCL/HA, PHBV/HA, PLDLA/Bioglass, PLLA/TCP, PEEK/HA, PLLA/Mg, PLLA/cHA	Relatively high temperature Best suited for polymers and polymer-ceramic composites Strong thermal bond Faster than 3DP Depowering required	[85, 201, 211, 212, 214, 227, 243]
	Selective laser melting	250–700 µm	PDLLA/TCP, Mg, Ti6Al4V, Zn, CoCrMo alloy, Ti-6Al-4V, Ti13Nb13Zr, Ti29Nb13Ta4.6Zr, Ti35Nb5Ta7Zr, Ti24Nb4Zr8Sn, Ti7.5Mo, Ti40Nb	High temperature process, otherwise similar to SLA Similar speed as SLS Depowering required	[209, 236, 238, 240, 249–251]
	Electron beam meeting	50–100 µm	Ti6Al4V	Highest temperature process (high power) Suitable for metallic materials Similar speed as SLS	[189, 234, 253, 254, 256, 257]
	Fused filament fabrication	100–150 µm	Polymers: PCL, PLA, PLGA, PEEK, PPF	Best suited for processing polymers Heat involved for partially melting the material Very common and widely adapted Requires support Can be paired with other processes	[186, 258–262]

Continued on next page ...

Process	Resolution	Materials	Properties	Ref.
Wet-spinning-based AM	123–789 µm	Polymers: PCL, PCL/HA, SPCL-Si	Chemical solvents involved, hence risk of residual toxicity present	[266, 267, 269, 271, 272]
Direct writing	50–850 µm	Composite: β-TCP/HA 45S5 Bioglass	Suitable for bioprinting Low-temperature application, hence friendly to drugs and biological growth factors Ideal for bioceramic and bioglass scaffolds	[195, 264, 275–278]
SLA	25–100 µm	Epoxy/HA, poly(trimethylene carbonate)/nHA, poly(ethylene glycol-co-depsipeptide) hydrogel	Requires photopolymers Challenge in balancing biological, chemical, and mechanical properties	
DLP	25–100 µm	(PPF)/(TiO2)/oxybenzophenone, fumaric acid monoethyl ester (FAME), end-functionalized PDLLA oligomers, N-vinyl-2-pyrrolidone	Polymer-based process High speed and resolution Good processing speed	[287, 330]
CDLP	25–100 µm			
Solution electrospinning	0.1–20 µm (fiber diameter)	Polymer: PCL, PGA Polymer composites: hyaluronic acid/collagen, PCL/PGA Polymer-ceramic composites: PCL/HA, PCL/HA/collagen, PCL/PLA/HA	Capable of producing nanofibers, with a large surface to volume ratio Commonly used for wound dressing Collected fiber mimics bone ECM Slow throughput Solvents involved, with risk of residual toxicity Difficult to control due to the whipping instabilities Fiber size and density too fine, not ideal for guiding bone cell growth Can be potentially adapted into AM	[301, 302, 324, 332–335]
Melt electrospinning	10–50 µm (fiber diameter)	PCL, PLA, PLLA, PLGA PLA/collagen, PCL/SrBG, PLA/collagen/HA	Bigger fiber diameter than SES Faster throughput	[18, 72, 106, 307, 316–319, 322, 323]

We generally noticed a need for more discussion on the clinical applicability of fabrication technologies reported in the literature. Except for a few clinical cases [336, 337], the vast majority of the literature is exploratory and mainly focuses on the technology of the fabrication technologies and materials for a *potential* application in bone TE. Only short-term evaluation of bone growth attributes such as cell attachment, proliferation, differentiation, and early-stage mineralization is often discussed. Full-scale CSBD repair cases with complete recovery and degradation profiles must be sufficiently covered. Comparative evaluations of scaffold performance are usually included in the reports but only in the context of evaluating different material and process parameters within a short time frame (30 days). Longer-term evaluation needs to be included and compared with different fabrication techniques. More noticeably, the significantly low number of clinical trials makes categorical evaluation of the clinical success of the manufacturing processes difficult. Out of 160 articles reviewed here on biological performances of bone scaffolds, only ∼20% were based on *in vivo* studies. The remaining ∼80% were *in vitro* studies on design, material, and effects of processing parameters on the cellular aspects of the scaffolds. Despite the sheer number of reports and ongoing activities, the field is still in its infancy, and successful implementations of these technologies in a clinical setting are eagerly awaited. Adapting newly emerging and exciting fabrication technologies is likely to happen in stages. The next wave of research will likely focus on scrutinizing emerging bone scaffolds in clinical settings.

Chapter 3

Background on Multi-body Dynamics Formulation Methods

This chapter briefly introduces the main theories and techniques employed in the theoretical formulations. The dynamic equations of motion of the melt-electrospun fibers are formulated by treating the fibers as multi-body systems. As discussed in the next chapter, the fiber segment hanging between the nozzle tip and the collector plate is discretized into smaller segments referred to as *beads*. These segments are treated as rigid bodies, serially connected with viscoelastic links.

The formulation starts by developing a set of unconstrained equations of motion for the beads. A number of different forces act on the beads, namely gravitational, electric field, Coulomb, and viscoelastic forces. Next, constraints representing the viscoelastic joints, the nozzle and collector plate motions, and the sticking/anchoring effect of the fiber on the collector plate are appended to the unconstrained equations to describe the dynamics of the fiber fully.

Several methods are available for formulating dynamic problems, including *Newtonian mechanics*, *Lagrangian mechanics*, *Hamiltonian mechanics*, *Kane's method*, and *Gibbs-Appell equations*. However, when dealing with multi-body systems involving a large number of rigid bodies, such as the fiber in question, Lagrangian mechanics and Kane's methods stand out because they can provide more systematic and efficient solutions. Kane's method, particularly, is advantageous for systems that involve many

constraints or complex constraint equations. Because of these advantages, Kane's method is selected to develop the unconstrained equations of motion.

The viscoelastic joint connecting the bodies, the prescribed nozzle and collector plate motions, and the sticking/anchoring effects observed when the fiber sticks to the collector plate reduce the degrees of freedom of the multi-body system. These restrictions are included in the dynamic formulation as constraint equations. An effective strategy is required to formulate and manage these constraints.

The *Udwadia-Kalaba* (UK) [338] method offers a unique approach for formulating constrained equations of motion. This method simplifies the mathematical formulation and improves the computational efficiency of the simulation by converting constraint equations into equivalent virtual constraint forces. These forces are added to the forces vector in the unconstrained equations of motion, resulting in the final fully constrained equations of motion.

Because of their unique advantages, Kane's and Udwadia-Kalaba's methods are combined in the current study. Kane's method is employed for generating the unconstrained equations of motion of the system, which will be comprised of a large number of beads. On the other hand, the UK method generates the constraint equations and combines them with the unconstrained equations to complete the problem formulation.

The resulting set of coupled algebraic differential equations (DAE) is numerically solved to determine the time histories of the individual beads. The solution is further post-processed to construct the fiber's time history by assembling the individual segments' time histories.

3.1 Kane's Method for Unconstrained Equations of Motion

The unconstrained equations of motion of the fiber are derived using *Kane's method*, which is a generalized approach based on Lagrange's Method and D'Alembert's principle[339–343]. The general form of the equations of motion using Kane's method is

given by the *dynamical equations*:

$$\boldsymbol{F}^* + \boldsymbol{F} = 0 \tag{3.1}$$

where $\boldsymbol{F^*}$ and $\boldsymbol{F}$ in Eq. (3.1) are vectors of the generalized inertia fores F_r^* and the generalized active forces F_r, respectively, that are calculated form:

$$F_r^* = \sum_{i=1}^{N} \boldsymbol{v}_r^i . \boldsymbol{R}_i^* \quad (r = 1, ..., \nu), \tag{3.2}$$

$$F_r = \sum_{i=1}^{N} \boldsymbol{v}_r^i . \boldsymbol{R}_i \quad (r = 1, ..., \nu), \tag{3.3}$$

where $\boldsymbol{v}_r^i$, $\boldsymbol{R}_i^*$, and $\boldsymbol{R}_i$ are the r^{th} partial velocity, the inertia force, and the resultant force on the i^{th} bead, respectively. N is the number of beads in the system, and ν the number of generalized coordinates, which is equal to $7N$. The partial velocities are calculated using the center of mass velocities.

In addition to the linear forces, torsional elastic and damping moments are included at joint locations in the model to influence and control the radius of curvature of the fiber. The effects of these moments are added to the model by introducing generalized inertia moments $\boldsymbol{T}^*$ and generalized active moments $\boldsymbol{T}$. The rotational aspect of the equation of motion is represented by:

$$\boldsymbol{T}^* + \boldsymbol{T} = 0. \tag{3.4}$$

$\boldsymbol{T}^*$ and $\boldsymbol{T}$ are calculated from equivalent formulas using the partial angular velocities of the beads. *i.e.*,

$$T_r^* = \sum_{i=1}^{N} \boldsymbol{\omega}_r^i . \boldsymbol{T}_i^*, \tag{3.5}$$

$$T_r = \sum_{i=1}^{N} \boldsymbol{\omega}_r^i . \boldsymbol{T}_i. \tag{3.6}$$

The unconstrained equations of motion of the multi-body system are given by combining the linear and rotational Kane's dynamical equations into a system of equations, *i.e.*,

$$\begin{bmatrix} \boldsymbol{F}^* + \boldsymbol{F} \\ \boldsymbol{T}^* + \boldsymbol{T} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.7}$$

Equation (3.7) could be rearranged in terms of the coefficients of $\boldsymbol{u}_i$, $\dot{\boldsymbol{u}}_i$, and $\ddot{\boldsymbol{u}}_i$, to yield the familiar matrix form of the equations of motion:

$$\boldsymbol{M}(\boldsymbol{u},t)\ddot{\boldsymbol{u}} + \boldsymbol{B}(\boldsymbol{u},t)\dot{\boldsymbol{u}} + \boldsymbol{K}(t)\boldsymbol{u} = \boldsymbol{\mathcal{F}}, \tag{3.8}$$

where $\boldsymbol{M}$, $\boldsymbol{B}$, $\boldsymbol{K}$, and $\boldsymbol{\mathcal{F}}$ are the system's mass matrix, damping coefficient matrix, stiffness matrix, and the externally applied forces, respectively.

Equation (3.8) can be further rearranged to furnish:

$$\boldsymbol{M}(\boldsymbol{u},t)\ddot{\boldsymbol{u}} = \boldsymbol{Q}(\boldsymbol{u},\dot{\boldsymbol{u}},t), \tag{3.9}$$

where $\boldsymbol{Q}$ is commonly known as the vector of *impressed forces*.

$$\boldsymbol{Q}(\boldsymbol{u},\dot{\boldsymbol{u}},t) = -\boldsymbol{B}(\boldsymbol{u},t)\dot{\boldsymbol{u}} - \boldsymbol{K}(t)\boldsymbol{u} + \boldsymbol{\mathcal{F}}. \tag{3.10}$$

Equation (3.9) is the system of unconstrained equations of motion of the multi-body system. When the constraint forces are ready, they are added to $\boldsymbol{Q}$ in this equation to arrive at the constrained equations of motion of the system.

3.2 Udwadia-Kalaba (UK) Method for Constrained Equations of Motion

Suppose the multi-body system has a vector of holonomic constraint equations $\boldsymbol{\varphi}$, expressed in terms of a vector of the independent generalized coordinates $\boldsymbol{u}$, *i.e.*,

$$\boldsymbol{\varphi}(\boldsymbol{u},t) = \boldsymbol{0}. \tag{3.11}$$

$\boldsymbol{\varphi}$ could be differentiated twice with respect to time to yield:

$$\frac{d^2}{dt^2}\boldsymbol{\varphi}(\boldsymbol{u},t) = \ddot{\boldsymbol{\varphi}}(\ddot{\boldsymbol{u}},\dot{\boldsymbol{u}},\boldsymbol{u},t) = \boldsymbol{0}. \tag{3.12}$$

Equation (3.12) could be rearranged into a matrix form:

$$\boldsymbol{A}(\dot{\boldsymbol{u}},\boldsymbol{u},t)\ddot{\boldsymbol{u}} = \boldsymbol{b}(\dot{\boldsymbol{u}},\boldsymbol{u},t), \tag{3.13}$$

where matrix $\boldsymbol{A}$ is the coefficient of the acceleration vector $\ddot{\boldsymbol{u}}$, and everything else is collected to the right-hand side of the equation and denoted by the vector $\boldsymbol{b}$.

The Udwadia-Kalaba method calculates the virtual constraint forces vector $\boldsymbol{Q}^c$ from Eq. (3.13) using the formula:

$$\boldsymbol{Q}^c = \boldsymbol{M}^{1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^{+}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}), \tag{3.14}$$

where the operator $(\,)^{+}$ is the Moore-Penrose inverse [344, 345].

Finally, the constrained equations of motion of the multi-body system are furnished by adding the virtual constraint forces vector $\boldsymbol{Q}^c$ to the vectors of impressed forces $\boldsymbol{Q}$ in Eq. (3.9), *i.e.*,

$$\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{Q} + \boldsymbol{Q}^c, \tag{3.15}$$

$$\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{Q} + \boldsymbol{M}^{1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^{+}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}). \tag{3.16}$$

3.3 Rotational Kinematics

Mathematically treating each bead as a rigid body allows one to include and investigate the effects of the rotational motion on the curvatures of the fiber. Similar models developed by other researchers, such as those presented in [315], mostly ignore the rotational components of the motions and approximate the problem as particle dynamics. In some cases, limits on the radius of the curvatures of the fiber were forced upon the models using techniques not necessarily related to the physics governing the motions of the beads. In addition to the rectilinear forces, the work presented here includes the effects of fiber bending resistances on motion dynamics.

However, including the rotational motion in the problem formulation is relatively complicated. It often requires an effective strategy to minimize complications in problem formulation and avoid potential pitfalls arising from singularities during the simulation.

Since unit-quaternions, compared to Euler angles, are generally less susceptible to singularity issues, they are adapted here as the primary generalized coordinates. However, because the components of unit-quaternions do not represent angles—neither in degrees nor in radians senses—they are often difficult to interpret and require conversion to other forms of rotation representations to communicate results. Therefore, while quaternions are used in the numerical integration, Euler angles and axis-angle

rotation representations are employed for handling initial conditions and inspection purposes. Below is a brief background discussion on these three rotation representations and their application in the forthcoming problem formulation.

3.3.1 Quaternion Algebra

Quaternions

Quaternions can be thought of as generalizations of complex numbers into 3D spaces. The complex number $z = x + y\mathbf{i} \,|\, z \in \mathbb{C}$ and $x, y \in \mathbb{R}$, with the property $\mathbf{i}^2 = -1$ has x and y as its real and imaginary parts, respectively, and $\mathbf{i}$ as the imaginary unit. Hamilton extended this notation in 1843 to cover complex numbers in 3D spaces, leading to a new numbering system we commonly refer to today as a quaternion (denoted by $\mathbb{H}$ after Hamilton). *i.e.*,

$$\boldsymbol{q} = w + a\mathbf{i} + b\mathbf{j} + c\mathbf{k}, \quad \boldsymbol{q} \in \mathbb{H}, \quad w, a, b, c \in \mathbb{R}, \tag{3.17}$$

where w in Eq. (3.17) is the real part of the quaternion $\boldsymbol{q}$, and a, b and c its imaginary parts. The $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the imaginary units in 3D space satisfying the properties:

$$\begin{gathered} \mathbf{i}^2 = \ \mathbf{j}^2 = \ \mathbf{k}^2 = \mathbf{i}\,\mathbf{j}\,\mathbf{k} = \ -1, \\ \mathbf{ij} = \mathbf{k}, \ \ \mathbf{jk} = \mathbf{i}, \ \ \mathbf{ki} = \mathbf{j}, \\ \mathbf{ik} = \ -\mathbf{j}, \ \ \mathbf{kj} = \ -\mathbf{i}, \ \ \mathbf{ji} = \ -\mathbf{k}. \end{gathered} \tag{3.18}$$

In engineering, a quaternion is usually represented as a four-element vector, made up of the real and imaginary parts without the imaginary $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ units included in the notations. However, algebraic operations defined on the quaternion vectors still obey the imaginary units multiplication rules described in Eq.(3.18), *i.e.*,

$$\boldsymbol{q} = \begin{bmatrix} w \\ \boldsymbol{v} \end{bmatrix}, \tag{3.19}$$

where $\boldsymbol{v}$ is a vector of the imaginary parts of the quaternion given by:

$$\boldsymbol{v} = \begin{bmatrix} b \\ c \\ d \end{bmatrix}. \tag{3.20}$$

Quaternion Norm

Given a quaternion $\boldsymbol{q}$ in a vector notation as in Eq. (3.17), its norm is calculated in the same manner as for a traditional three-component vector. *i.e.*,

$$\|\boldsymbol{q}\| = \sqrt{w^2 + \|\boldsymbol{v}\|^2} = \sqrt{w^2 + a^2 + b^2 + c^2}. \tag{3.21}$$

Unit-quaternions are special quaternions whose norm is 1. *i.e.*,

$$\|\boldsymbol{q}\| = w^2 + a^2 + b^2 + c^2 = 1. \tag{3.22}$$

As will be shown shortly in upcoming sections, unit-quaternions play an essential role in representing rotations and orientations of rigid bodies.

Quaternion Conjugate

The conjugate of a quaternion $\boldsymbol{q}$, denoted by $\boldsymbol{q}^*$, is another quaternion given by:

$$\boldsymbol{q}^* = \begin{bmatrix} w \\ -\boldsymbol{v} \end{bmatrix}. \tag{3.23}$$

Quaternion Inverse

The inverse of a quaternion $\boldsymbol{q}$ is another quaternion given by the expression:

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{\|\boldsymbol{q}\|^2} = \frac{1}{w^2 + \|\boldsymbol{v}\|^2} \begin{bmatrix} w \\ -\boldsymbol{v} \end{bmatrix}. \tag{3.24}$$

Since the norm of a unit quaternion is 1, its inverse is:

$$\boldsymbol{q}^{-1} = \boldsymbol{q}^* = \begin{bmatrix} w \\ -\boldsymbol{v} \end{bmatrix}. \tag{3.25}$$

Quaternion Addition/Subtraction

Given two quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in vector forms:

$$\boldsymbol{q}_1 = \begin{bmatrix} w_1 \\ \boldsymbol{v}_1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{q}_2 = \begin{bmatrix} w_2 \\ \boldsymbol{v}_2 \end{bmatrix},$$

Their sum/difference is:

$$\boldsymbol{q}_1 \pm \boldsymbol{q}_2 = \begin{bmatrix} w_1 & \pm & w_2 \\ \boldsymbol{v}_1 & \pm & \boldsymbol{v}_2 \end{bmatrix}. \tag{3.26}$$

Quaternion Multiplication (Hamilton Product) $\otimes$

The product (also known as the Hamilton product) of two quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is a quaternion given by:

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} w_1 w_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 \\ w_1 \boldsymbol{v}_2 + w_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_1 \end{bmatrix}. \tag{3.27}$$

Quaternion product is not commutative. *i.e.*, $\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \neq \boldsymbol{q}_2 \otimes \boldsymbol{q}_1$.

3.3.2 Orientation Representations

In the dynamic analysis of multi-body systems, it is often required to describe the orientation of a rigid body in space relative to another reference frame. As shown in Fig. 3.1, this is usually done by attaching a coordinate system to the rigid body such that the origin of this coordinate system coincides with some point of interest on the rigid body (such as the center of mass or a joint). Additionally, the three axes are selected to parallel the body's principal dimensions. Since such a frame is fixed on the body, the body's orientation in space can be described mathematically by the relationships between the body-fixed and the reference coordinate systems.

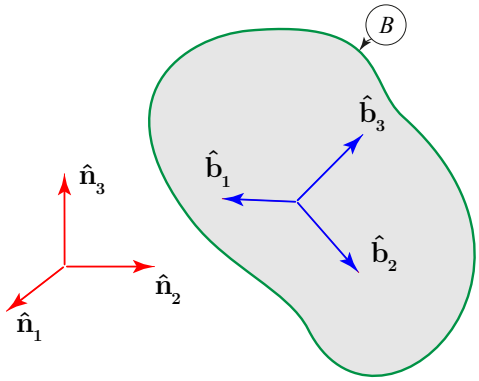


Figure 3.1: Body-fixed frame

The body frame is assumed to be initially parallel to the reference frame before it is brought to its current orientation through some successive rotations. Below are three methods widely used to describe such rotations.

Axis-Angle Rotation Representation

The axis-angle method describes the orientation of a given rigid body as a single rotation of the reference coordinate system by an angle θ about a unit-vector $\hat{\boldsymbol{e}}$ that passes through the origin (Fig. 3.2). This rotation is represented by:

$$\boldsymbol{\theta} = \theta\hat{\boldsymbol{e}} \quad \text{or} \quad Rot : (\hat{\boldsymbol{e}}, \theta). \tag{3.28}$$

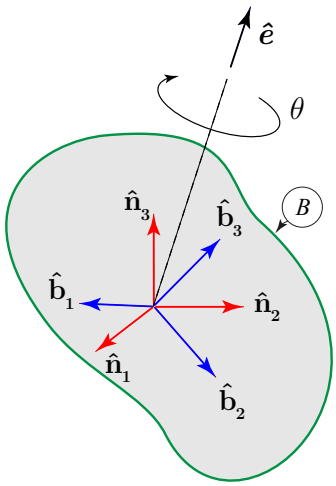


Figure 3.2: Axis-angle rotation representation

Euler Angles Representation

In Euler angles orientation representation (Fig. 3.3), the orientation of the body-fixed frame relative to the reference frame is achieved by three successive rotations of the reference frame about the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ coordinate axes – but not necessarily in that particular order. In fact, different rotation sequences most likely lead to different outcomes, and care must be taken in specifying the sequences.

The rotations can be taken about the axes of the original reference coordinate system $(\hat{\boldsymbol{n}}_1, \hat{\boldsymbol{n}}_2, \hat{\boldsymbol{n}}_3)$ *(extrinsic rotations)* or the axes of the intermediate rotated coordinate systems $(\hat{\boldsymbol{x}}', \hat{\boldsymbol{y}}', \hat{\boldsymbol{z}}')$ and $(\hat{\boldsymbol{x}}", \hat{\boldsymbol{y}}", \hat{\boldsymbol{z}}")$ *(intrinsic rotations.)* Unless and otherwise stated explicitly, the discussion presented in the current work follows intrinsic rotations.

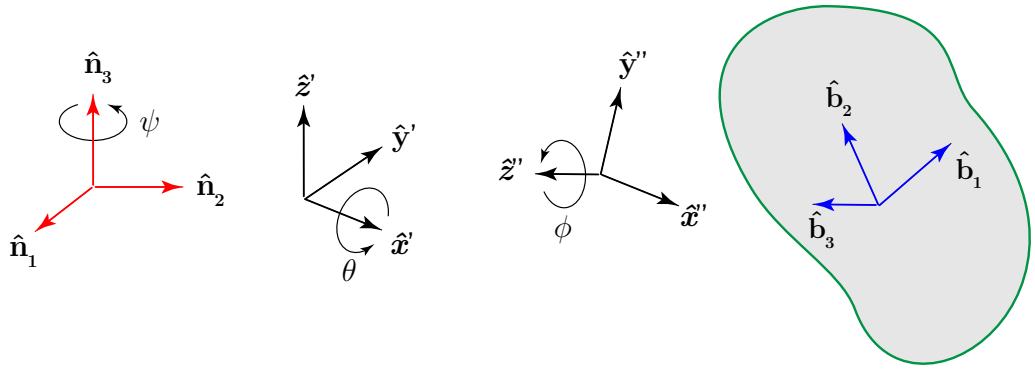


Figure 3.3: Rotation representation using Euler angles

The three angles of rotations in Fig. 3.3 are known as Euler angles. Three Euler angles denote the final orientation of the rigid body with a side note on the sequences of rotation followed. For example, the rotation:

$$Euler\ angles = (\psi, \theta, \phi)\,, \quad \hat{\boldsymbol{z}}\hat{\boldsymbol{x}}'\hat{\boldsymbol{z}}''$$

indicates a first rotation about the $\hat{\boldsymbol{z}}$ axis by angle ψ, followed by a rotation about the $\hat{\boldsymbol{x}}'$ axis by angle θ, and a final rotation about the $\hat{\boldsymbol{z}}''$ axis by angle ϕ.

With the obvious restriction that no two consecutive rotations can be about the same axis, there could be twelve possible rotation sequences to choose from. *i.e.*,

$$\begin{aligned}
&\hat{\boldsymbol{x}} \to \hat{\boldsymbol{y}} \to \hat{\boldsymbol{x}}, \quad && \hat{\boldsymbol{x}} \to \hat{\boldsymbol{z}} \to \hat{\boldsymbol{x}}, \\
&\hat{\boldsymbol{y}} \to \hat{\boldsymbol{x}} \to \hat{\boldsymbol{y}}, \quad && \hat{\boldsymbol{y}} \to \hat{\boldsymbol{z}} \to \hat{\boldsymbol{y}}, \\
&\hat{\boldsymbol{z}} \to \hat{\boldsymbol{x}} \to \hat{\boldsymbol{z}}, \quad && \hat{\boldsymbol{z}} \to \hat{\boldsymbol{y}} \to \hat{\boldsymbol{z}}
\end{aligned} \tag{3.29}$$

$$\begin{aligned}
&\hat{\boldsymbol{x}} \to \hat{\boldsymbol{y}} \to \hat{\boldsymbol{z}}, \quad && \hat{\boldsymbol{x}} \to \hat{\boldsymbol{z}} \to \hat{\boldsymbol{y}}, \\
&\hat{\boldsymbol{y}} \to \hat{\boldsymbol{x}} \to \hat{\boldsymbol{z}}, \quad && \hat{\boldsymbol{y}} \to \hat{\boldsymbol{z}} \to \hat{\boldsymbol{x}}, \\
&\hat{\boldsymbol{z}} \to \hat{\boldsymbol{x}} \to \hat{\boldsymbol{y}}, \quad && \hat{\boldsymbol{z}} \to \hat{\boldsymbol{y}} \to \hat{\boldsymbol{x}}.
\end{aligned} \tag{3.30}$$

The first group of six rotations, in which the first and the last rotations are always about the same axis, is known as *proper Euler angles*. The rotations in which all three axes are involved in the rotation are known as the *Tait-Brian angles*.

The Euler angle representation is the most widely used due to its intuitiveness and long history in aircraft navigational systems. However, there are well documents drawbacks related to singularity issues (gimbal lock) arising from the presence of

trigonometric terms, making this method less ideal for the simulation of problems such as the present work. Therefore, its current use is limited to only entering initial values and extracting simulation results for inspection purposes.

Euler Parameters (Unit Quaternions)

Given an axis-angle representation of rotation of a rigid body by an angle θ about a unit vector axis $\hat{\boldsymbol{n}}$, the equivalent Euler parameters representation is calculated from:

$$\hat{\boldsymbol{e}} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\ \hat{\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2) \begin{bmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{bmatrix} \end{bmatrix} \tag{3.31}$$

This method of rotation representation addresses the singularity issue of Euler angles by adding one more redundant parameter. Euler parameters, however, are less intuitive because the values of the parameters do not represent angles. This makes them very tricky to interpret, and they often require conversion back to Euler angles or other rotation descriptions. Since the orientation of a rigid body can be sufficiently described by using only three parameters (because pure rotation is a three-degree of freedom system), the four Euler parameters are not entirely independent. In fact, a constraint equation is required to enforce this observation to avoid redundancy.

Since $\hat{\boldsymbol{n}}$ in Eq. (3.31) is a unit vector representing the axis of rotation, the norm of $\hat{\boldsymbol{e}}$ calculated from:

$$\|\hat{\boldsymbol{e}}\|^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2 = cos^2(\theta/2) + sin^2(\theta/2)\,\|\hat{\boldsymbol{n}}\|^2 \tag{3.32}$$

is 1.

This brings to evidence that Euler parameters are, in fact, unit quaternions, and they do obey and satisfy all conditions and operations defined over the quaternion space $\mathbb{H}$. In addition, Eq. (3.32) is the constraint equation required to bring the number of degrees of freedom of systems described by Euler parameters back to three, consistent with other rotation representation methods. Now that the connection between unit quaternions and Euler parameters is clarified, throughout the remainder of the text, Euler parameters will be represented by the four-element unit quaternion

vector notation as:

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.33}$$

Matrix Representation of Unit Quaternions

Unit quaternions could be used as linear mapping operators to transform or rotate vectors from one coordinate system to another. Below are defined two important matrices, linear in the quaternion elements. They will be used frequently in upcoming discussions and the theoretical problem formulation in Chapter 4.

$$\boldsymbol{E} = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \tag{3.34}$$

$$\boldsymbol{G} = \begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{3.35}$$

Coordinate Transformations and Rotations

Two scenarios of interest exist where the previously discussed orientation descriptions could play a critical role.

1. Coordinate Transformation: Given a vectorial representation of a physical quantity in the bases of a coordinate system, we are interested in expressing this vector in terms of the bases of another coordinate system. In this case, the physical quantity and its vector representation remain unchanged. The vector could be considered a fixed line in space. The goal is to express this vector using different bases. This is analogous to describing an event in different languages—the event remains the same, but the language used to describe it could be different.

2. Rotation: In the second scenario, the vector is rotated (the physical quantity it represents is now altered) relative to all frames of reference and points in a different direction. The description of the rotated vector in any given set of bases differs from the original vector because the old and the new vectors represent two different physical quantities. In this case, the change in physical quantity is equally observed in all coordinate systems involved regardless of the choice of bases.

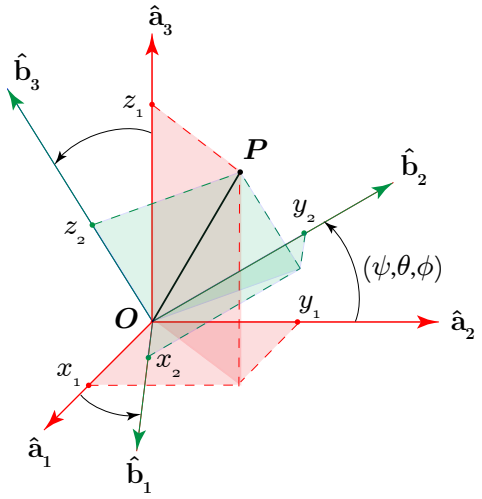


Figure 3.4: Coordinate transformation

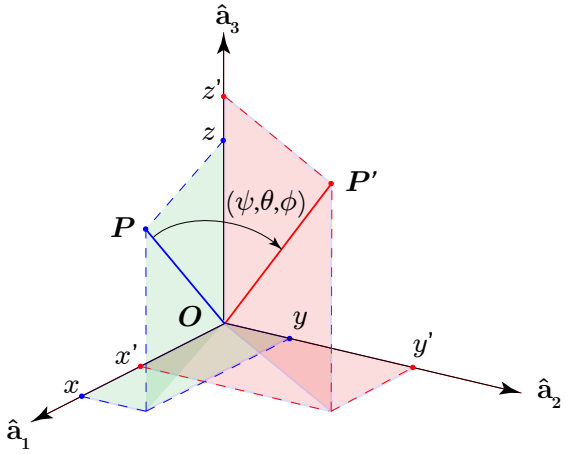


Figure 3.5: Rotation of a vector

Knowledge of the relative orientations between the coordinate systems or between the original and rotated vectors could be exploited to construct linear mapping functions that elegantly handle these two scenarios.

Rotation matrix

a) From Euler Angles

If the current orientation of a coordinate system is achieved by successively rotating a given reference coordinate system by three Euler angles ψ, θ and ϕ about the intrinsic $\hat{\boldsymbol{z}}$, $\hat{\boldsymbol{y}}'$, and $\hat{\boldsymbol{x}}''$ axes, respectively, the rotation matrices for the three rotations are given in matrix form by:

$$\boldsymbol{R}\left(\hat{\boldsymbol{z}},\psi\right) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.36}$$

$$\boldsymbol{R}\left(\hat{\boldsymbol{y}}',\theta\right) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{3.37}$$

$$\boldsymbol{R}\left(\hat{\boldsymbol{x}}'',\phi\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}. \tag{3.38}$$

The resultant rotation matrix between the reference frame and the final frame is then calculated from the product of these three matrices, *i.e.*,

$$\boldsymbol{R}(\psi,\theta,\phi) = \boldsymbol{R}\left(\hat{\boldsymbol{z}},\psi\right)\boldsymbol{R}\left(\hat{\boldsymbol{y}}',\theta\right)\boldsymbol{R}\left(\hat{\boldsymbol{x}}'',\phi\right), \tag{3.39}$$

$$\boldsymbol{R}(\psi,\theta,\phi) = \begin{bmatrix} \mathrm{c}\,\psi\,\mathrm{c}\,\theta & \mathrm{c}\,\psi\,\mathrm{s}\,\phi\,\mathrm{s}\,\theta - \mathrm{c}\,\phi\,\mathrm{s}\,\psi & \mathrm{s}\,\phi\,\mathrm{s}\,\psi + \mathrm{c}\,\phi\,\mathrm{c}\,\psi\,\mathrm{s}\,\theta \\ \mathrm{c}\,\theta\,\mathrm{s}\,\psi & \mathrm{c}\,\phi\,\mathrm{c}\,\psi + \mathrm{s}\,\phi\,\mathrm{s}\,\psi\,\mathrm{s}\,\theta & \mathrm{c}\,\phi\,\mathrm{s}\,\psi\,\mathrm{s}\,\theta - \mathrm{c}\,\psi\,\mathrm{s}\,\phi \\ -\mathrm{s}\,\theta & \mathrm{c}\,\theta\,\mathrm{s}\,\phi & \mathrm{c}\,\phi\,\mathrm{c}\,\theta \end{bmatrix}. \tag{3.40}$$

b) From Axis-Angle

A more general form the rotation matrix R is given in Eq. (3.41) for an axis-angle rotation $Rot : (\hat{\boldsymbol{n}},\gamma)$

$$\boldsymbol{R}(\hat{\boldsymbol{n}},\gamma) = \begin{bmatrix} \mathrm{c}\,\gamma + n_x^2(1-\mathrm{c}\,\gamma) & n_x n_y(1-\mathrm{c}\,\gamma) - n_z\,\mathrm{s}\,\gamma & n_x n_z(1-\mathrm{c}\,\gamma) + n_y\,\mathrm{s}\,\gamma \\ n_x n_y(1-\mathrm{c}\,\gamma) + n_z\,\mathrm{s}\,\gamma & \mathrm{c}\,\gamma + n_y^2(1-\mathrm{c}\,\gamma) & n_y n_z(1-\mathrm{c}\,\gamma) - n_x\,\mathrm{s}\,\gamma \\ n_x n_z(1-\mathrm{c}\,\gamma) - n_y\,\mathrm{s}\,\gamma & n_y n_z(1-\mathrm{c}\,\gamma) + n_x\,\mathrm{s}\,\gamma & \mathrm{c}\,\gamma + n_z^2(1-\mathrm{c}\,\gamma) \end{bmatrix}. \tag{3.41}$$

c) From Unit Quaternions

The rotation matrix can also be determined from the unit quaternion representation of orientation between the two frames using the formula:

$$\boldsymbol{R} = \boldsymbol{G}\boldsymbol{E}^T, \tag{3.42}$$

where $\boldsymbol{E}$ and $\boldsymbol{G}$ are the quaternion matrices described in Eq. (3.34) and (3.35), respectively. The product furnishes:

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_0q_1 + 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{3.43}$$

Coordinate transformation using $\boldsymbol{R}$

If $\boldsymbol{R}$ is a rotation matrix rotating Frame 1 with bases $\hat{\mathbf{b}}_1 : (\hat{\mathbf{x}}_1,\ \hat{\mathbf{y}}_1,\ \hat{\mathbf{z}}_1)$ to a final orientation designated by Frame 2 with bases $\hat{\mathbf{b}}_2 : (\hat{\mathbf{x}}_2,\ \hat{\mathbf{y}}_2,\ \hat{\mathbf{z}}_2)$, the relationships between the bases is given by:

$$\hat{\mathbf{b}}_1 = \boldsymbol{R}\ \hat{\mathbf{b}}_2, \qquad \hat{\mathbf{b}}_2 = \boldsymbol{R}^T\ \hat{\mathbf{b}}_1. \tag{3.44}$$

Vector Projection with $\boldsymbol{R}$

Alternatively, $\boldsymbol{R}$ could project a given vector onto another coordinate system. For example, if a position vector $\boldsymbol{r}$ is given by $\boldsymbol{r}_1.\hat{\mathbf{b}}_1$ in Frame 1 and by $\boldsymbol{r}_2.\hat{\mathbf{b}}_2$ in Frame 2, the relationships between the two projections are given by:

$$\boldsymbol{r}_1 = \boldsymbol{R}\ \boldsymbol{r}_2, \qquad \boldsymbol{r}_2 = \boldsymbol{R}^T\ \boldsymbol{r}_1. \tag{3.45}$$

Note that $\boldsymbol{r}$ is not rotated, just bases translations!

Rotation of a vector using $\boldsymbol{R}$

The rotation matrix $\boldsymbol{R}$ may also rotate a vector by some angles within a given coordinate system. If $\boldsymbol{v}_1.\hat{\mathbf{b}}$ is the original vector, and it is rotated by the same Euler angles and sequences of rotations that led to the formation of $\boldsymbol{R}$, then the rotated vector $\boldsymbol{v}_2.\hat{\mathbf{b}}$ (notice the same coordinate system/bases) can be calculated from:

$$\boldsymbol{v}_2.\hat{\mathbf{b}} = \boldsymbol{R}^T\boldsymbol{v}_1.\hat{\mathbf{b}}. \tag{3.46}$$

Coordinate Rotation with Unit Quaternions

Instead of rotation matrices, an equivalent unit quaternion $\boldsymbol{q}$ may also be used to do the coordinate rotation in Eq. (3.44) and arrive at the same result. However, all three-element vectors must first be converted into quaternions by using the vectors as the imaginary part of the quaternions and assigning zero values to the real parts. *i.e.*,

$$\begin{bmatrix} 0 \\ \hat{\mathbf{b}}_1 \end{bmatrix} = \boldsymbol{q}^* \otimes \begin{bmatrix} 0 \\ \hat{\mathbf{b}}_2 \end{bmatrix} \otimes \boldsymbol{q}, \quad or \quad \begin{bmatrix} 0 \\ \hat{\mathbf{b}}_2 \end{bmatrix} = \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \hat{\mathbf{b}}_1 \end{bmatrix} \otimes \boldsymbol{q}^*, \tag{3.47}$$

where $\otimes$ is the quaternion product operator and $\boldsymbol{q}^*$ the conjugate of $\boldsymbol{q}$.

Similarly, the projection of vector $\mathbf{r}$ in Eq. (3.45) can be handled using unit quaternions. *i.e.*,

$$\begin{bmatrix} 0 \\ \mathbf{r}_1 \end{bmatrix} = \boldsymbol{q}^* \otimes \begin{bmatrix} 0 \\ \mathbf{r}_2 \end{bmatrix} \otimes \boldsymbol{q}, \quad or \quad \begin{bmatrix} 0 \\ \mathbf{r}_2 \end{bmatrix} = \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \mathbf{r}_1 \end{bmatrix} \otimes \boldsymbol{q}^*. \tag{3.48}$$

Rotation of a vector with Unit-Quaternions

The vector rotation of Eq. (3.46) can also be performed using unit quaternions:

$$\begin{bmatrix} 0 \\ \boldsymbol{v}_2 \end{bmatrix} = \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}_1 \end{bmatrix} \otimes \boldsymbol{q}^*. \tag{3.49}$$

Successive Rotations

If a given coordinate system was rotated using successive rotation matrices $\boldsymbol{R}_1$, $\boldsymbol{R}_2$, ..., $\boldsymbol{R}_N$ to get to its current orientation, the equivalent rotation matrix could be calculated by multiplying the individual rotation matrices in the same order. *i.e.*,

$$\boldsymbol{R} = \boldsymbol{R}_1 \boldsymbol{R}_2 ... \boldsymbol{R}_N. \tag{3.50}$$

If unit quaternions were used in each rotation instead of rotation matrices, the equivalent unit-quaternion could be found by multiplying the individual quaternions.

$$\boldsymbol{q} = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \otimes ... \otimes \boldsymbol{q}_N. \tag{3.51}$$

Conversion between Euler Angles and Quaternions

A set of Euler angles could be converted to an equivalent unit quaternion dependent on the sequence of the Euler rotations. Each Euler rotation can be represented by an

axis-angle description and easily be converted to a quaternion form using Eq. (3.31). Since the three rotations happen sequentially, the equivalent quaternion representing the three Euler angles could be calculated using the successive quaternion formula in Eq. (3.51). For example, for the $z-y-x$ rotation sequence, the Euler angles ψ, θ and ϕ can be represented in axis-angle notations:

$$\begin{aligned} &Rot1 : ([0\ 0\ 1]^T, \psi), \\ &Rot2 : ([0\ 1\ 0]^T, \theta), \\ &Rot3 : ([1\ 0\ 0]^T, \phi). \end{aligned} \tag{3.52}$$

The unit quaternions corresponding to each rotation are:

$$\boldsymbol{q}(\psi) = \begin{bmatrix} \cos(\psi/2) \\ \sin(\psi/2) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}, \quad \boldsymbol{q}(\theta) = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{bmatrix}, \quad \boldsymbol{q}(\phi) = \begin{bmatrix} \cos\phi/2 \\ \sin(\phi/2) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}. \tag{3.53}$$

As per Eq. (3.51), the resultant quaternions are given by the Hamilton product of the three quaternions in the same order. *i.e.*,

$$\boldsymbol{q} = \boldsymbol{q}(\psi) \otimes \boldsymbol{q}(\theta) \otimes \boldsymbol{q}(\phi). \tag{3.54}$$

3.3.3 Quaternion Kinematics

Angular velocity

Angular velocities of rigid bodies are required to describe the dynamic behavior of the rigid bodies adequately. Suppose the unit quaternion describing the orientation of a rigid body and its first-time derivative are known. In that case, the angular velocity of the rigid body can be described in the inertial and the body-fixed coordinates from the following two formulas:

$$\boldsymbol{\omega}^{\mathbf{b}} = 2\dot{\boldsymbol{G}}\boldsymbol{q} = \ 2\boldsymbol{G}\dot{\boldsymbol{q}}, \tag{3.55}$$

$$\boldsymbol{\omega}^{\mathbf{n}} = 2\dot{\boldsymbol{E}}\boldsymbol{q} = \ 2\boldsymbol{E}\dot{\boldsymbol{q}}, \tag{3.56}$$

where $\boldsymbol{E}$ and $\boldsymbol{G}$ are the quaternion matrices given by Eq. (3.34) and (3.35).

Chapter 4

Theoretical Model Development

Chapter Abstract

A novel analytical model development and simulation of the melt-electrospinning (MES) process is presented. Unconstrained equations of motion for describing a discretized melt-electrospun fiber were formulated using Kane's method. The motions of the spinneret and the collector plate were also incorporated into the kinematics formulation to simulate direct-writing and three-dimensional (3D) printing scenarios. Constraints describing viscoelastic joints in the system and the phenomenon of the melt-electrospun fiber adhering to the collector plate were implemented using the Udwadia Kalaba method. Rotational viscoelastic elements are introduced to mimic the dampening of the whipping motion observed in the unstable region. A novel algorithm was devised to simulate continuously with no limit on the total duration and phase transitions. System responses such as fiber diameter, collection size, fiber elongations, and jet speeds were monitored for changes in control parameters, including applied voltage, collector distance, and flow rate. The results demonstrated close agreement with experimental observations in the literature. The jet started relatively straight on the onset of its trajectory and developed a subtle whipping motion as it advanced towards the collector plate. Reduction in the amplitude of the whipping motion was observed once the free end of

the fiber adhered to the collector plate.

4.1 Introduction

There has been a growing interest in adapting MES for biomedical applications [19, 20]. This is mainly because toxic chemicals, primarily associated with solvents used in solution-electrospinning (SES), are absent in MES, making the latter an ideal fabrication method for applications such as wound dressing and tissue engineering [15–18]. The close resemblance between the intra-fiber geometry of melt-electrospun fibers and the micro-architecture of the extracellular matrix (ECM) of bone tissue is another factor that makes MES a potential candidate for the fabrication of bone scaffolds [2]. However, the convergence of MES and 3D manufacturing processes requires a thorough understanding of the interdependencies among process, material, and output parameters. For a given set of some of these processing and material parameters, our primary interest in the proposed work is to develop a theoretical model for predicting the motion profiles of the melt-electrospun fiber as it exits the spinneret tip and accumulates onto a flat collector plate. Knowledge of this allows one to predict important morphological and topological output attributes. These include the electrospun fiber diameter, collection pattern, collection density, and other parameters that may be of practical interest when adapting MES as a direct-writing or a 3D fabrication process [279].

To this end, numerous experimental and analytical models have been proposed in the past by several groups. These models were used to investigate different characteristics of the electrospun fiber and to understand the process's behavior better. Most consequentially of all, Taylor [346], in 1964, formulated an empirical relationship between the applied voltage and the shape of a charged droplet. He estimated the critical voltage required to trigger the jet formation in SES in terms of the material properties and some geometrical parameters in the system. Based on the work in [346], other groups [314, 347] have proposed additional mathematical models to describe the geometrical characteristics of the stable jet and the bending instabilities regions. One example is a model proposed by He et al.[348] to predict the critical length of the jet, below which the jet remains stable and relatively straight (see

Fig. 4.1.a). Electrospinning configuration in which the collector distance is set shorter than this critical length is commonly known as near-field electrospinning (NFES). The bending instabilities under such a configuration are noticeably absent, resulting in a well-aligned fiber topology that is easily controlled by the motion of the spinneret relative to the collector plate [349, 350]. Compared to SES, MES (see Fig. 4.1.b) is distinctly characterized by significantly dampened whipping motions. Dalton and Brown [279, 351] have experimentally investigated the effects of the processing parameters and the transverse speed of the collector plate on fiber alignments in MES. Their experiments revealed that the faster the relative speed of the collector plate with respect to the nozzle, the straighter the topology of the collected fiber and the thinner the average fiber diameter. The fiber tends to coil upon itself at zero or low relative speeds between the nozzle and the collector plate, as shown in Fig. 4.2.a [30]. Increasing the speed tends to open the coil and stretch the fiber in the direction of nozzle motion (Fig. 4.2.b). Another experimental study by Zheng et al.[352] has shown similar results. Using an NFES configuration for a given set of material parameters and processing conditions, they identified the collector moving speed (CMS) ranges in which straight, wavy, and coiled fiber depositions occur.

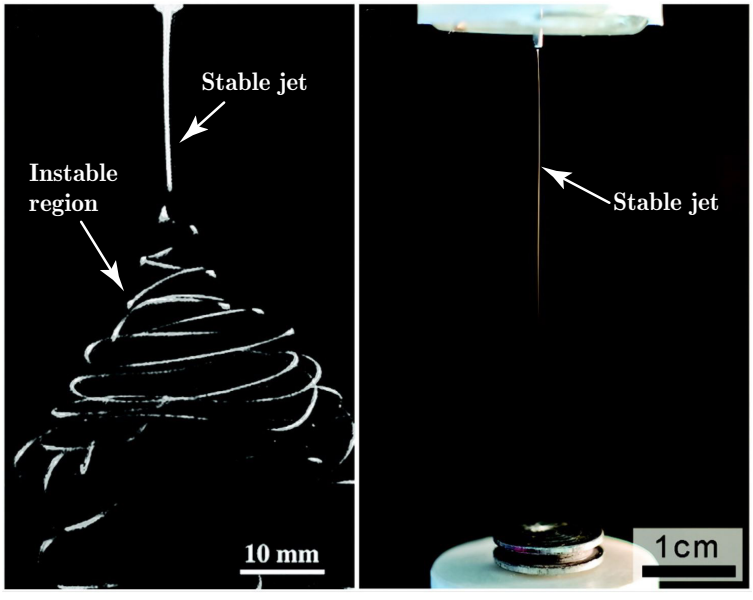


Figure 4.1: a) Solution-based electrospinning, b) MES. Reproduced with permission[30]

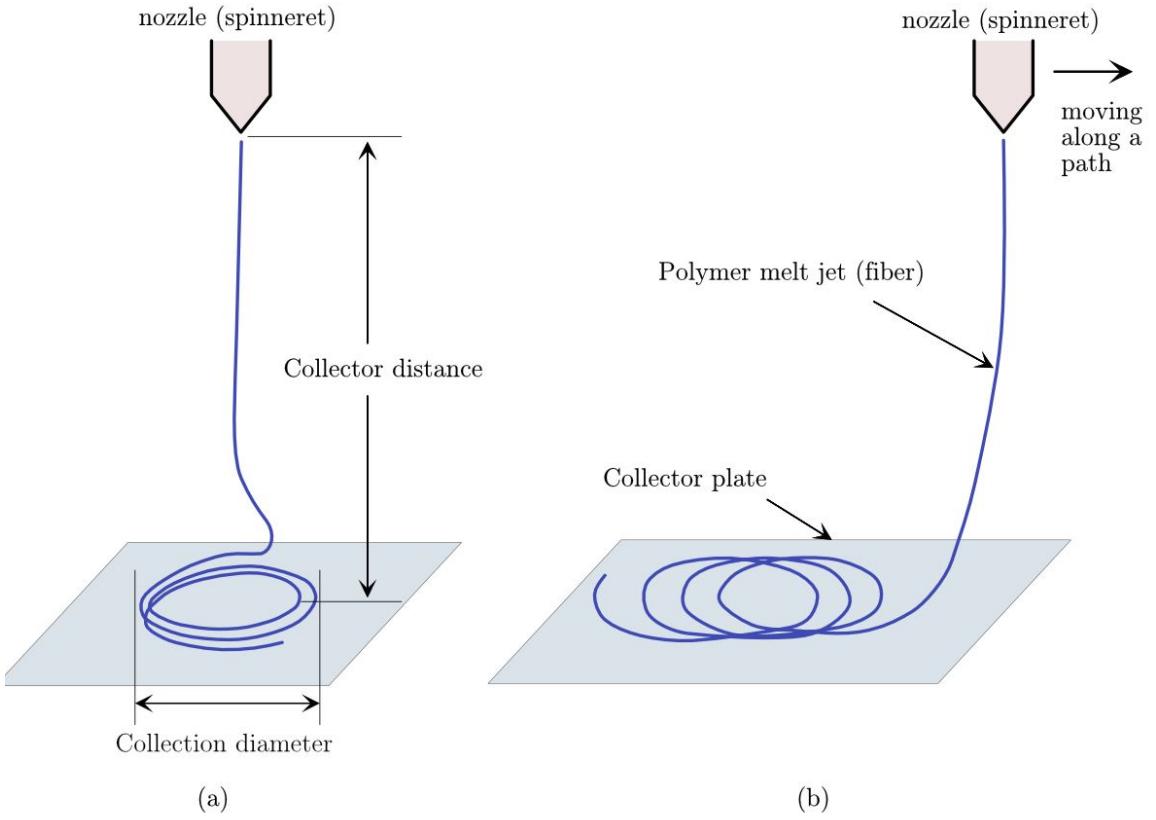


Figure 4.2: Schematic diagram of MES: a) with no relative velocity between the nozzle and collector plate, b) with a relative velocity between nozzle and collector plate. (adapted from [30])

Highly erratic and chaotic motions often characterize the bending instabilities region, making theoretical model development and validation challenging tasks [297]. Reneker et al. [315] developed the first dimensionless bending instability model for SES. They treated the active fiber segment as a series of beads connected via simple Maxwell viscoelastic links (*i.e.*, a spring connected in series with a dashpot). They simulated the motions of the beads subjected to gravitational, surface tension, Coulomb, electric field, and viscoelastic forces. Other models were later developed [297, 298, 353–357] to investigate the effects of the process parameters (such as applied voltage, collector distance, and melting temperature) on fiber morphology and topology.

However, most of the models mentioned above were developed explicitly for SES and were based on oversimplified assumptions. For instance, as demonstrated by Costa and Ribeiro [28], it usually takes 10 to 15 layers of a generalized Maxwell viscoelastic rheological model to curve-fit experimental rheology test data with reasonable accuracy. Hence, using single-layer Maxwell viscoelastic links in the above models is suspected to result in sub-optimal representations of the creep and relaxation behaviors of the materials involved [29]. The electrospinning models reviewed here also implicitly assume that one end of the fiber would always remain free in the air without touching down on the collector plate. As a result of this assumption, the anchoring effects are mostly ignored. This is potentially problematic, especially in the simulation of NFES, where the fiber often buckles under the support of the collector plate [30]. Additionally, if the spinneret and the collector plate were to have independent motions (as is commonly the case in 3D printing scenarios), external momentum would be imparted to the fiber by the moving platforms, altering its dynamic behavior. Discussions regarding the effects of platform motions on fiber dynamics are generally absent in the literature.

In the present study, we propose a theoretical model based on Reneker's work [315] that addresses the deficiencies mentioned above and introduces several additional features and improvements. Specifically, the novelties of the proposed work are: 1) the anchoring and whipping dampening forces generated as a result of the fiber sticking on the collector plate are included in the formulation; 2) the motions of the spinneret

and the collector plate are considered to allow simulation of direct-writing and 3D printing scenarios; 3) Maxwell-based standard solid (SLS) rheological links are used as viscoelastic joints for improved representation of creep and relaxation behaviors of the molten polymer; 4) rotational viscoelastic elements are incorporated into the model to capture the whipping motion dampening phenomenon; 5) the rotational degrees of freedoms of the discretized fiber segments are included in the dynamic formulation, and 6) a computationally efficient algorithm is developed for continuous simulation of MES processes without the need to define the model size ahead of time.

4.2 Model Development

To mathematically model the dynamic behavior of the melt-electrospun fiber, the *active fiber length*—the segment between the spinneret tip and the collector plate, not including the fiber that would have already been accumulated on the collector plate—is discretized into smaller segments. These segments, interchangeably referred to as *links* or *beads*, are treated as rigid bodies with their own inertial, electrical, and material properties. Successive segments in the system are connected via the SLS joint to capture the viscoelastic nature of the polymer melt.

The dynamic equations of the electrospun fiber are formulated by first developing a set of unconstrained equations of motions for the fiber segments using Kane's method [340, 342, 358]. Holonomic constraints arise in the system from the viscoelastic joints and the fiber segments sticking on the collector plate. These constraints are imposed on the unconstrained equations using the Udwadia-Kalaba method [338, 359–362]. The resulting set of coupled differential-algebraic equations (DAE) is numerically integrated to determine the time histories of the individual fiber segments. The results are subsequently assembled to construct the time history of the whole electrospun fiber. The surface tension forces, known to play dominant roles in SES, are only considered during the initial stages of the process. They are intentionally excluded once the jet is initiated because their effects are negligible compared to the other forces after the jet is initiated [363].

4.2.1 Generalized Coordinates

The first step in the modeling process is to select a set of generalized coordinates describing the positions and orientations of each link in the system. We have chosen to represent the positions of the fiber segments (*beads*) by the absolute Cartesian coordinates of their center of masses (see Fig. 4.3). Although Euler angles are one of the most widely used methods for orientation representation due to their intuitiveness, we have opted for quaternions (Euler parameters) in the present work. This decision is taken to avoid singularity and Gimbal-lock issues associated with the former. These issues arise from the presence of trigonometric terms in the Euler angles, making the method less ideal for the simulation of large-displacement problems such as the present work [364]. As a result, the use of Euler angles in this work is limited to entering initial values and extracting simulation results for inspection purposes only. Quaternions address the singularity issue by adding one more redundant parameter [365–370]. However, they are less intuitive because the parameters do not represent angles in the usual sense. This lack of intuitiveness makes them very tricky to interpret and often requires conversion back to Euler angles or other rotation description methods.

Let $\boldsymbol{r}_i(t)$ and $\boldsymbol{q}_i(t)$ be the position and orientation of bead $\mathcal{B}_i$ in 3D Euclidean space. Written in matrix form:

$$\boldsymbol{r}_i(t) = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{4.1}$$

$$\boldsymbol{q}_i(t) = \begin{bmatrix} q_{i0} & q_{i1} & q_{i2} & q_{i3} \end{bmatrix}^T, \tag{4.2}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{q}_i$ can be concatenated to form a single vector of the generalized coordinates $\boldsymbol{u}_i$; *i.e.*,

$$\boldsymbol{u}_i(t) = \begin{bmatrix} \boldsymbol{r}_i(t)^T \;\middle|\; \boldsymbol{q}_i(t)^T \end{bmatrix}^T = \begin{bmatrix} x_i & y_i & z_i & q_{i0} & q_{i1} & q_{i2} & q_{i3} \end{bmatrix}^T. \tag{4.3}$$

Although only six coordinates suffice to fully describe the position and orientation of an unconstrained rigid body (*i.e.*, three for translation and three for rotation), using unit quaternions in the above definition introduces an extra dependent coordinate per bead. The constraint equation required to reduce the degree of freedom back to six

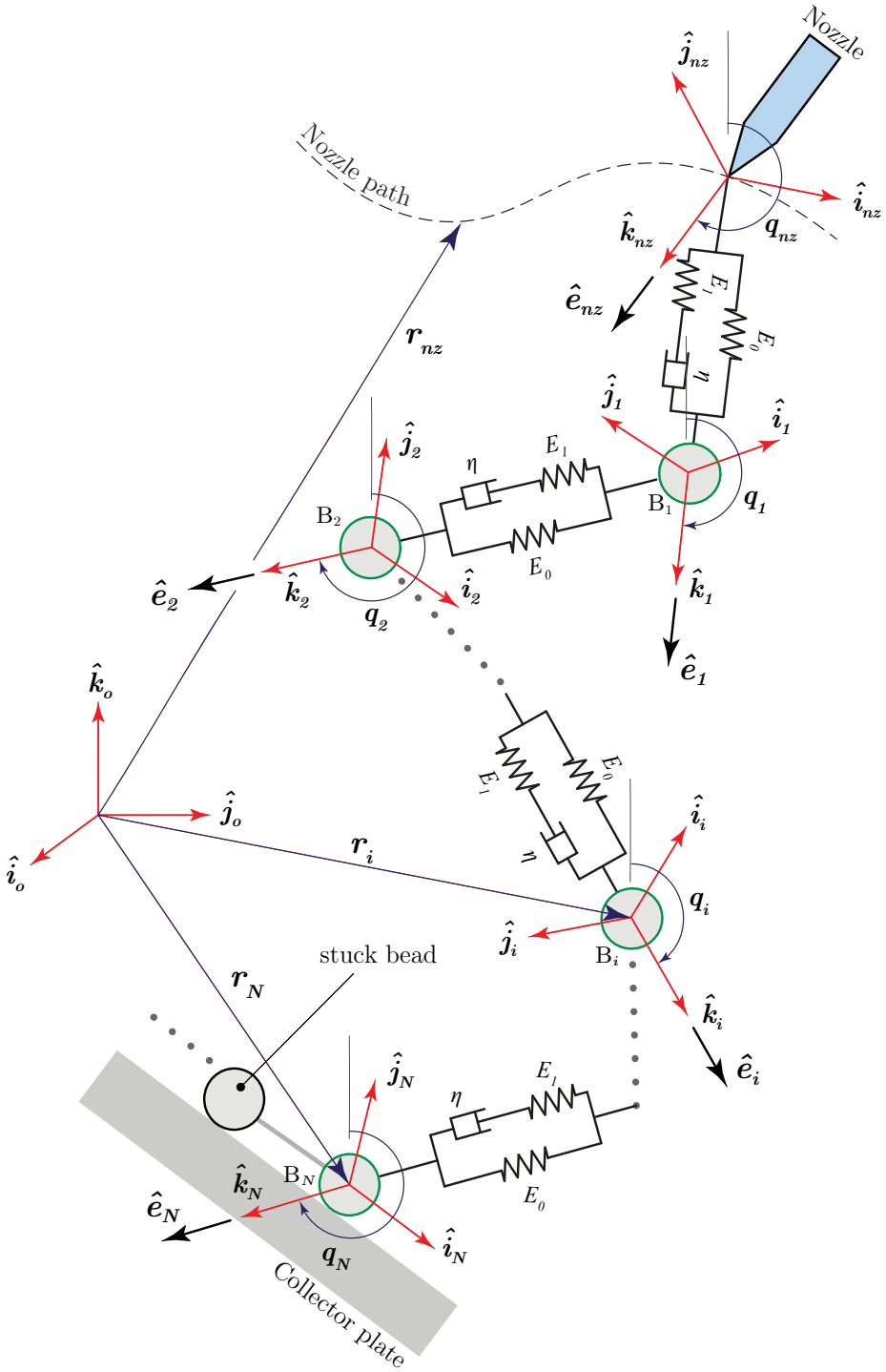


Figure 4.3: Melt-electrospun fiber as a viscoelastic chain

comes from the fact that the norm of the unit quaternion is 1 [371]; *i.e.*,

$$\|\boldsymbol{q}_i\| = q_{i0}^2 + q_{i1}^2 + q_{i2}^2 + q_{i3}^2 = 1. \tag{4.4}$$

The generalized velocities vector $\dot{\boldsymbol{u}}_i(t)$ is obtained by taking the total time derivative of the generalized coordinates vector $\boldsymbol{u}_i(t)$ in Eq. (4.3); *i.e.*,

$$\dot{\boldsymbol{u}}_i(t) = \frac{d\boldsymbol{u}_i(t)}{dt} = \begin{bmatrix} \dot{x}_i & \dot{y}_i & \dot{z}_i & \dot{q}_{i0} & \dot{q}_{i1} & \dot{q}_{i2} & \dot{q}_{i3} \end{bmatrix}^T. \tag{4.5}$$

4.2.2 Partial Linear Velocities and Partial Angular Velocities

Denoting the linear and angular velocities of the i^{th} bead by $\boldsymbol{v}_i$ and $\boldsymbol{\omega}_i$, respectively, its partial linear velocity $\bar{\boldsymbol{v}}_i$ and partial angular velocity $\bar{\boldsymbol{\omega}}_i$ are defined such that [340]:

$$\boldsymbol{v}_i \triangleq \sum_{r=0}^{\nu} \bar{v}_{ir}\dot{u}_{ir} + \bar{\boldsymbol{v}}_{it}, \tag{4.6}$$

$$\boldsymbol{\omega}_i \triangleq \sum_{r=0}^{\nu} \bar{\omega}_{ir}\dot{u}_{ir} + \bar{\boldsymbol{\omega}}_{it}, \tag{4.7}$$

where $\bar{v}_{ir}$, $\bar{\omega}_{ir}$, and $\dot{u}_{ir}$ are, respectively, the r^{th} components of $\bar{\boldsymbol{v}}_i$, $\bar{\boldsymbol{\omega}}_i$, and the generalized velocity vector $\dot{\boldsymbol{u}}_i$. Equivalently, suppressing r over the range of the generalized coordinates in the system, the above may be rewritten in the following compact matrix from:

$$\boldsymbol{v}_i = \bar{\boldsymbol{v}}_i\dot{\boldsymbol{u}}_i + \bar{\boldsymbol{v}}_{it}, \tag{4.8}$$

$$\boldsymbol{\omega}_i = \bar{\boldsymbol{\omega}}_i\dot{\boldsymbol{u}}_i + \bar{\boldsymbol{\omega}}_{it}. \tag{4.9}$$

The bead velocities $\boldsymbol{v}_i$ and $\boldsymbol{\omega}_i$, however, can be directly computed using the generalized coordinates and the generalized velocities from:

$$\boldsymbol{v}_i = \dot{\boldsymbol{r}}_i, \tag{4.10}$$

$$\boldsymbol{\omega}_i = 2\boldsymbol{G}_i\dot{\boldsymbol{q}}_i, \tag{4.11}$$

where $\boldsymbol{G}_i$ is a linear mapping matrix defined in terms of the components of the unit quaternion $\boldsymbol{q}_i$ as:

$$\boldsymbol{G}_i(\boldsymbol{q}_i) \triangleq \begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{4.12}$$

A quick inspection of Eq. (4.8) and Eq. (4.10) reveals that the partial linear velocity matrix $\bar{\boldsymbol{v}}_i$ in Eq. (4.8) is simply the coefficient matrix of $\dot{\boldsymbol{r}}$ in Eq. (4.10). *i.e.*,

$$\bar{\boldsymbol{v}}_i = \boldsymbol{I}_3, \tag{4.13}$$

where $\boldsymbol{I}_3$ is a 3-by-3 identity matrix.

Similarly from Eq. (4.9) and (4.11), the partial angular velocity $\bar{\boldsymbol{\omega}}_i$ is

$$\bar{\boldsymbol{\omega}}_i = 2\boldsymbol{G}_i. \tag{4.14}$$

4.2.3 Admissible Forces and Torques

Gravitational Force

The force of gravity exerted on the i^{th} fiber segment of mass m_b is:

$$\boldsymbol{f}_{Gi} = m_b \boldsymbol{g}, \tag{4.15}$$

where $\boldsymbol{g}$ is the acceleration due to gravity expressed in the inertial frame of reference; *i.e.*,

$$\boldsymbol{g} = [0 \quad 0 \quad -9.81]^T \mathrm{m\,s^{-2}}. \tag{4.16}$$

As reported by Wunner et al. [372], once the jet is initiated, the gravitational force has a negligible effect on the jet path, with the effect of the force even further diminishing with increasing flow rate. However, the gravitational force contribution is still included in the model formulation because it is expected to play a dominant role when the applied voltage is below the critical limit and there is a positive flow rate pumping material out of the spinneret. Under this scenario, no fiber thinning would be observed because of the insufficient voltage, but the material acts as an extruded molten polymer.

Electric Field Force

In MES, an electrically charged molten polymer is pulled towards a grounded collector plate under the actions of electric field forces. The resulting electric field between the spinneret and the collector platform is generally non-uniformly distributed. It may be expressed as a function of position (*i.e.*, $\boldsymbol{E}(\boldsymbol{r})$) following similar procedures discussed

in [373]. However, in the present work, the spatial distribution of $\boldsymbol{E}$ is assumed to be uniform for clear and concise demonstration. The presumed uniformity of $\boldsymbol{E}$ suffices to demonstrate the important aspects of the proposed model. In addition, all beads in the system are assumed to carry identical electrical charges q, the magnitude of which is determined based on the dielectric properties of the molten polymer, melt flow rate, and the size of the discretized fiber segment. The uniform electric filed vector $\boldsymbol{E}$, hence is approximated by the formula:

$$\boldsymbol{E} = \frac{V}{h_c}\ \hat{\boldsymbol{e}}_E, \tag{4.17}$$

where V is the applied potential difference between the spinneret and the collector plate, and h_c is the collector distance. $\hat{\boldsymbol{e}}_E$, which is an equipment parameter, is a unit vector in the direction of the electric field (same as the direction of spinning). For example, a vertical MES setup will have: $\hat{\boldsymbol{e}}_E = [0\ 0\ -1]^T$.

In the case of a bead idealized as a charged particle, the electric field force exerted on it is determined from:

$$\boldsymbol{f}_{Ei} = q\boldsymbol{E} = q\frac{V}{h_c}\ \hat{\boldsymbol{e}}_E. \tag{4.18}$$

Coulomb Forces

Since all beads in the system are assumed to carry identical electrical charges, q, they also apply repulsive forces on each other. The net Coulomb force on the i^{th} bead is given by the sum of all Coulomb forces exerted by the other $N-1$ beads; *i.e.*,

$$\boldsymbol{f}_{Ci} = k_e\ q^2\ \sum_{\nu=1}^{N}\left(\frac{1}{\|\Delta\boldsymbol{r}_\nu\|^2}\right)\hat{\boldsymbol{e}}_\nu, \quad (\forall\ \nu \neq i), \tag{4.19}$$

where k_e is the Coulomb constant ($= 8.987 \times 10^9$ N.m^2/C^2), $\Delta\boldsymbol{r}_\nu$ the position vector pointing from bead $\mathcal{B}_\nu$ to bead $\mathcal{B}_i$, and $\hat{\boldsymbol{e}}_\nu$ is the associated unit vector defined by:

$$\Delta\boldsymbol{r}_\nu = \boldsymbol{r}_i - \boldsymbol{r}_\nu = \begin{bmatrix} x_i - x_\nu \\ y_i - y_\nu \\ z_i - z_\nu \end{bmatrix}, \quad \hat{\boldsymbol{e}}_\nu = \frac{\Delta\boldsymbol{r}_\nu}{\|\Delta\boldsymbol{r}_\nu\|}. \tag{4.20}$$

Substitution of Eq. (4.20) into Eq. (4.19) furnishes:

$$\boldsymbol{f}_{Ci} = k_e\ q^2\ \sum_{\nu=1}^{N}\left(\frac{\Delta\boldsymbol{r}_\nu}{\|\Delta\boldsymbol{r}_\nu\|^3}\right), \quad (\forall\ \nu \neq i). \tag{4.21}$$

Viscoelastic Forces and Torques

Viscoelasticity is a material property in which the material simultaneously exhibits both viscous and elastic deformation characteristics in response to changes in applied stresses [374]. Upon application or removal of the stresses, the elastic component of the material undergoes instantaneous changes in strain (*i.e.*, elastic strain). Meanwhile, the viscous component of the material deforms over a period of time. The molten polymer jet in MES shares these characteristics. It deforms in a time-dependent manner in response to changes in stresses induced by the other forces in the system. This behavior manifests itself as additional resistance forces internal to the material that attempt to delay the resulting changes in strain.

Viscoelasticity is usually modeled using springs and dashpots arranged in series, parallel, or a combination of these configurations. The springs and the dashpots represent the elastic and viscous aspects of the material, respectively [375, 376]. The generalized Maxwell standard linear solid (SLS) viscoelastic rheological model is one such model (Fig. 4.4). This model has a spring connected in parallel to one or more layers of Maxwell links (a spring connected in series to a dashpot). Compared to other models[363], it allows the inclusion of as many layers as needed for proper fine-tuning and estimation of the elastic parameters (E_0, E_1, η_1, E_2, η_2, ...), and hence closely representing the experimentally observed material behavior. However, to reduce computational load during simulation, only the first layer is included in the present study (Fig. 4.5). The governing constitutive equation relating the stress, strain, and elastic parameters in the Maxwell SLS model is given by [377]:

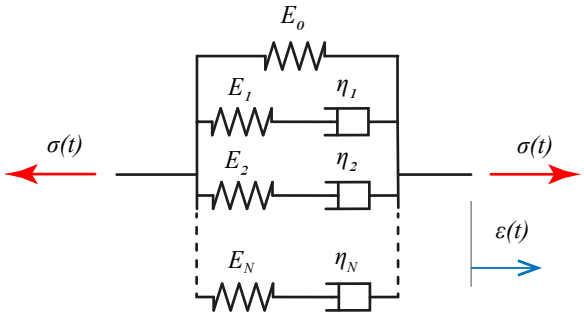


Figure 4.4: Generalized Maxwell-based standard solid (SLS) model

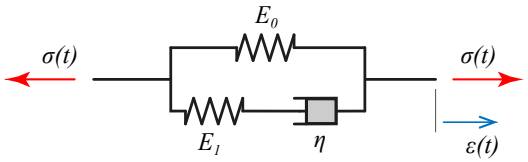


Figure 4.5: A single-layer standard linear solid (SLS) model

$$\sigma(t) + \frac{\eta}{E_1(t)}\dot{\sigma}(t) = E_0(t)\ \varepsilon(t) + \frac{\eta\ (E_0(t) + E_1(t))}{E_1(t)}\ \dot{\varepsilon}(t). \tag{4.22}$$

The space between the spinneret tip and the collector plate is typically characterized by a non-uniform temperature gradient that starts with the melt temperature T_m at the spinneret tip and gradually drops down to the ambient temperature T_{amb} at the surface of the collector plate. The values of the elastic constants E_0, E_1, and η, hence, change in response to this change in temperature. Constructing the complete temperature dependency profiles of these parameters or their response behavior is beyond the scope of the present study. However, to acknowledge the phenomena, these parameters are represented in the current study as linear functions of T_0 and T_{amb} and the bead's position relative to the spinneret tip, which is a function of time itself, *i.e.*,

$$E_0(t) = E_0'(T_0, T_{amb}, \boldsymbol{r}_i(t)), \qquad E_1(t) = E_1'(T_0, T_{amb}, \boldsymbol{r}_i(t)), \qquad \eta(t) = \eta'(T_0, T_{amb}, \boldsymbol{r}_i(t)), \tag{4.23}$$

where E_0', E_1', and η' are the values of the elastic constants measured at T_m.

For the efficient formation of the algebraic differential equations, the stress $\sigma(t)$ is selected as one of the independent variables in the present work. Each joint connecting subsequent beads has its own stress variable denoted by $\sigma_i(t)$. The strain terms, on the other hand, are calculated from changes in the initial link length; hence they are dependent on the generalized coordinates and generalized velocities of the beads, *i.e.*,

$$\varepsilon_i(t) = \frac{\|\Delta\boldsymbol{r}_i\| - l_0}{l_0}, \tag{4.24}$$

where l_0 is the initial unstretched link length, and $\Delta\boldsymbol{r}_i$ the i^{th} *displacement vector* calculated from the position vectors of beads $\mathcal{B}_i$ and $\mathcal{B}_{i-1}$ as:

$$\Delta\boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{r}_{i-1} = \begin{bmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \\ z_i - z_{i-1} \end{bmatrix}. \tag{4.25}$$

E_0, E_1, and η in Eq. (4.22) are the elastic parameters of the SLS model. They are estimated from rheological tests following procedures outlined in Reference [28]. For a given tensile stress value $\sigma_i(t)$, the magnitude of the axial viscoelastic force $\boldsymbol{f}_{Vi}$ in the fiber is calculated by simply multiplying $\sigma_i(t)$ by the instantaneous cross-sectional area of the fiber; *i.e.*,

$$\|\boldsymbol{f}_{Vi}\| = \sigma_i(t)\left(\frac{\pi d_i^2}{4}\right), \tag{4.26}$$

where d_i is the fiber diameter measured at the i^{th} link. Because bead $\mathcal{B}_i$ is connected with beads $\mathcal{B}_{i-1}$ and $\mathcal{B}_{i+1}$ using two different viscoelastic joints, the net viscoelastic force on $\mathcal{B}_i$ is given by the vector sum of the reaction viscoelastic forces contributed by the other two beads:

$$\boldsymbol{f}_{Vi} = \boldsymbol{f}_{Vi-} + \boldsymbol{f}_{Vi+}, \tag{4.27}$$

where $\boldsymbol{f}_{Vi-}$ and $\boldsymbol{f}_{Vi+}$ are the viscoelastic force contributions due to the bead's interactions with beads $\mathcal{B}_{i-1}$ and $\mathcal{B}_{i+1}$, respectively. Using Eq. (4.26), these become:

$$\boldsymbol{f}_{Vi-} = -\sigma_i(t)\left(\frac{\pi d_i^2}{4}\right)\hat{\boldsymbol{e}}_i, \qquad \boldsymbol{f}_{Vi+} = \sigma_{i+1}(t)\left(\frac{\pi d_{i+1}^2}{4}\right)\hat{\boldsymbol{e}}_{i+1}. \tag{4.28}$$

The unit vectors $\hat{\boldsymbol{e}}_i$ and $\hat{\boldsymbol{e}}_{i+1}$, pointing in the directions of their respective displacement vectors, are calculated from:

$$\hat{\boldsymbol{e}}_i = \frac{\Delta\boldsymbol{r}_i}{\|\Delta\boldsymbol{r}_i\|}, \qquad \hat{\boldsymbol{e}}_{i+1} = \frac{\Delta\boldsymbol{r}_{i+1}}{\|\Delta\boldsymbol{r}_{i+1}\|}. \tag{4.29}$$

Equation (4.28) – (4.29) may then be substituted into Eq. (4.27) to yield an expression for the net viscoelastic force applied on the i^{th} bead:

$$\boldsymbol{f}_{Vi} = -\sigma_i(t)\left(\frac{\pi d_i^2}{4}\right)\frac{\Delta\boldsymbol{r}_i}{\|\Delta\boldsymbol{r}_i\|} + \sigma_{i+1}(t)\left(\frac{\pi d_{i+1}^2}{4}\right)\frac{\Delta\boldsymbol{r}_{i+1}}{\|\Delta\boldsymbol{r}_{i+1}\|}. \tag{4.30}$$

In the absence of any bending resistance or limiting/dampening mechanism, a link element in the system could swing excessively relative to its neighboring links, creating sharp bends at and in the vicinity of each joint. In practice, melt electrospun fibers have some bending resistance, as evidenced by the smooth radii of curvatures of the collected fibers. In the proposed model, we introduce rotational viscoelastic elements (see, Fig. 4.6) at the joints to account for the bending resistance of fibers

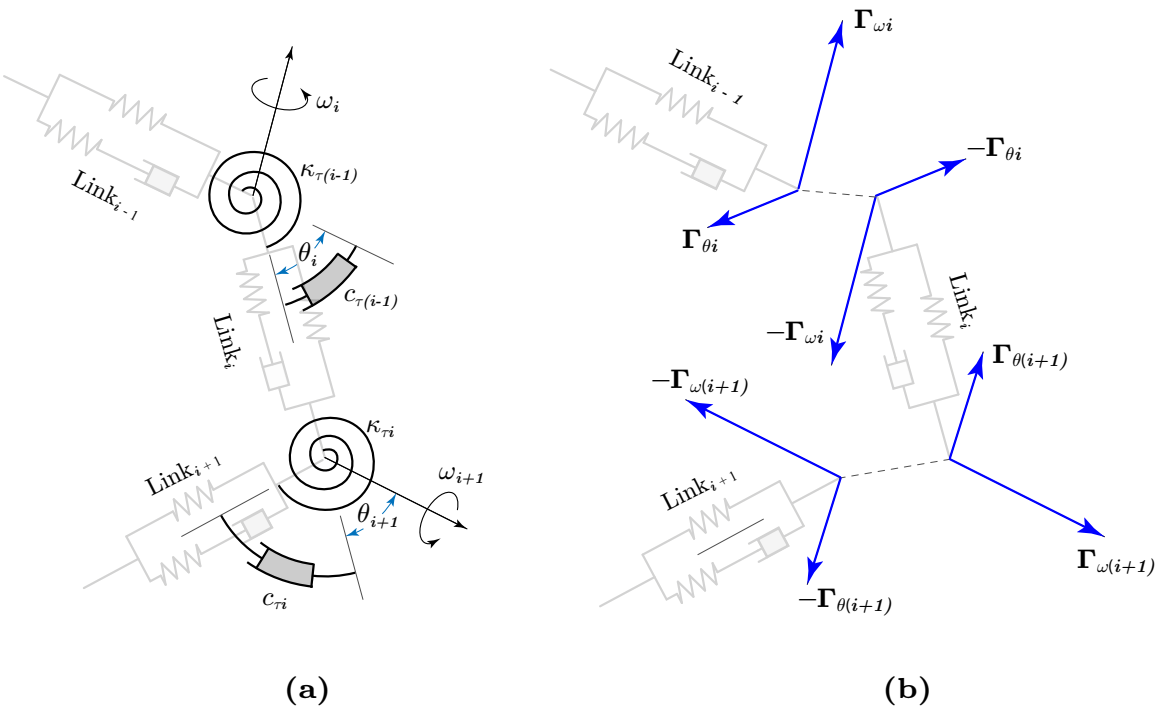


Figure 4.6: Joint elastic and damping bending resistance forces: a) Spring-dashpot system, b) Free body diagram

and to achieve a more realistic representation of smooth transitions from one link to another.

The torsional spring and the dashpot in the i^{th} link shown in Fig. 4.6 generate elastic and damping reaction torques between beads $\mathcal{B}_{i-1}$ and $\mathcal{B}_i$ proportional to the included angle θ_i and its time derivative $\dot{\theta}_i$, respectively. Since the spring and the dashpot are connected in parallel, the principle of superposition remains valid, and the torques they generate may be added together.

Hence, the reaction torque exerted by $\mathcal{B}_{i-1}$ on $\mathcal{B}_i$, denoted by $\boldsymbol{\Gamma}_{i-}$, is:

$$\boldsymbol{\Gamma}_{i-} = -\boldsymbol{\Gamma}_{\theta i} - \boldsymbol{\Gamma}_{\omega_i}, \tag{4.31}$$

$$\boldsymbol{\Gamma}_{i-} = -\kappa_{\tau i}\theta_i \; \hat{\boldsymbol{e}}_{\theta i} - c_{\tau i}\dot{\theta}_i \; \hat{\boldsymbol{e}}_{\omega i}, \tag{4.32}$$

where $\kappa_{\tau i}$ and $c_{\tau i}$ are the i^{th} bead's torsional spring stiffness and rotational damping coefficient, respectively. Similarly, the expression of the reaction torque on $\mathcal{B}_i$ due to

its interaction with $\mathcal{B}_{i+1}$, is obtained from:

$$\boldsymbol{\Gamma}_{i+} = \boldsymbol{\Gamma}_{\theta(i+1)} + \boldsymbol{\Gamma}_{\omega(i+1)}, \tag{4.33}$$

$$\boldsymbol{\Gamma}_{i+} = \kappa_{\tau(i+1)}\theta_{(i+1)}\ \hat{\boldsymbol{e}}_{\theta(i+1)} + c_{\tau(i+1)}\dot{\theta}_{(i+1)}\ \hat{\boldsymbol{e}}_{\omega(i+1)}. \tag{4.34}$$

The $\dot{\theta}\ \hat{\boldsymbol{e}}_{\omega}$ terms in Eq. (4.32) and Eq. (4.34) represent relative angular velocities between successive links, and may be calculated as:

$$\dot{\theta}_i\ \hat{\boldsymbol{e}}_{\omega i} = \Delta\boldsymbol{\omega}_i = \boldsymbol{\omega}_i - \boldsymbol{\omega}_{i-1}, \tag{4.35}$$

$$\dot{\theta}_{(i+1)}\ \hat{\boldsymbol{e}}_{\omega(i+1)} = \Delta\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_{i+1} - \boldsymbol{\omega}_i. \tag{4.36}$$

The out-of-plane unit normal vector $\hat{\boldsymbol{e}}_{\theta}$ and the associated angle θ are calculated from the generalized coordinates and the generalized velocities of beads $\mathcal{B}_{i-1}$, $\mathcal{B}_i$, and $\mathcal{B}_{i+1}$ as follows:

$$\hat{\boldsymbol{e}}_{\theta i} = \frac{\hat{\boldsymbol{e}}_{i-1} \times \hat{\boldsymbol{e}}_i}{\|\hat{\boldsymbol{e}}_{i-1} \times \hat{\boldsymbol{e}}_i\|}, \qquad \hat{\boldsymbol{e}}_{\theta(i+1)} = \frac{\hat{\boldsymbol{e}}_i \times \hat{\boldsymbol{e}}_{i+1}}{\|\hat{\boldsymbol{e}}_i \times \hat{\boldsymbol{e}}_{i+1}\|}, \tag{4.37}$$

$$\theta_i = \text{atan}(\|\hat{\boldsymbol{e}}_{i-1} \times \hat{\boldsymbol{e}}_i\|,\ \hat{\boldsymbol{e}}_{i-1} \cdot \hat{\boldsymbol{e}}_i), \tag{4.38}$$

$$\theta_{i+1} = \text{atan}(\|\hat{\boldsymbol{e}}_i \times \hat{\boldsymbol{e}}_{i+1}\|,\ \hat{\boldsymbol{e}}_i \cdot \hat{\boldsymbol{e}}_{i+1}), \tag{4.39}$$

where $\hat{\boldsymbol{e}}_i$ and $\hat{\boldsymbol{e}}_{i-1}$ are the displacement unit vectors of $\mathcal{B}_{i-1}$ and $\mathcal{B}_i$, respectively, as defined in Eq. (4.29). The relative angular velocity terms can also be reformulated using the generalized coordinates as (refer, Eq. (4.11):

$$\Delta\boldsymbol{\omega}_i = \boldsymbol{\omega}_i - \boldsymbol{\omega}_{i-1} = 2(\boldsymbol{G}_i\dot{\boldsymbol{q}}_i - \boldsymbol{G}_{i-1}\dot{\boldsymbol{q}}_{i-1}), \tag{4.40}$$

$$\Delta\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_{i+1} - \boldsymbol{\omega}_i = 2(\boldsymbol{G}_{i+1}\dot{\boldsymbol{q}}_{i+1} - \boldsymbol{G}_i\dot{\boldsymbol{q}}_i). \tag{4.41}$$

Lastly, the net torque applied on $\mathcal{B}_i$ is determined by summing together the reaction torques exerted by $\mathcal{B}_{i-1}$ and $\mathcal{B}_{i+1}$ (Eq. (4.32) and Eq. (4.34).

$$\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_{i-} + \boldsymbol{\Gamma}_{i+} \tag{4.42}$$

$$\boldsymbol{\Gamma}_i = \kappa_{\tau(i+1)}\theta_{(i+1)}\ \hat{\boldsymbol{e}}_{\theta(i+1)} + c_{\tau(i+1)}\Delta\boldsymbol{\omega}_{i+1} - \kappa_{\tau i}\theta_i\ \hat{\boldsymbol{e}}_{\theta i} - c_{\tau i}\Delta\boldsymbol{\omega}_i \tag{4.43}$$

Surface Tension Forces

At the onset of the MES process, a charged droplet is formed at the tip of the spinneret. Dominant surface tension forces initially characterize this droplet. However,

as the applied voltage increases, the electric field forces become stronger, forming a Taylor cone at the tip of the spinneret [346]. Once the applied voltage reaches a critical limit V_C, the electric field forces overcome the surface tension and initiate a charged polymer jet. The surface tension forces tend to pull the bead adjacent to the spinneret towards the center of the cone. The surface tension forces contributed by the remaining beads are not considered because, once the polymer jet is initiated, the magnitudes of the forces become negligible relative to other forces in the system [363, 378]. Such an assumption may not be valid in the case of SES, where the surface tension forces remain significant even after the jet is initiated. The direction of the surface tension force is opposite to the resultant of all other forces acting on the bead. However, for the sake of modeling simplicity, this direction is assumed to be opposite to the electric field vector $\hat{\boldsymbol{e}}_E$ defined in Equation (4.17). The magnitude is estimated from the formula $f_{CAP_1} = 2\gamma D$, where γ is the surface tension of the molten polymer, and D is the internal diameter of the spinneret tip in; *i.e.*,

$$\boldsymbol{f}_{CAP_i} = \begin{cases} -2\gamma D\ \hat{\boldsymbol{e}}_E, & (i = 1) \\ 0, & (i > 1), \end{cases} \tag{4.44}$$

4.2.4 Generalized Forces and Torques

The net external force on the i^{th} bead, denoted by $\boldsymbol{f}_i$, is calculated by summing the gravitational, electric field, Coulomb, viscoelastic, and surface tension forces.

$$\boldsymbol{f}_i = \boldsymbol{f}_{Gi} + \boldsymbol{f}_{Ei} + \boldsymbol{f}_{Ci} + \boldsymbol{f}_{Vi} + \boldsymbol{f}_{CAP_i}. \tag{4.45}$$

The corresponding *generalized active force*, $\boldsymbol{f}_i$, is then given by the product:

$$\boldsymbol{F}_i \triangleq \sum \bar{\boldsymbol{v}}_i \circ \boldsymbol{f}_i, \tag{4.46}$$

or equivalently,

$$\boldsymbol{F}_i = \bar{\boldsymbol{v}}_i^T \boldsymbol{f}_i = \boldsymbol{I}_3 \boldsymbol{f}_i, \tag{4.47}$$

where $\bar{\boldsymbol{v}}_i$ is the partial linear velocity of the bead defined in Eq. (4.13).

$\boldsymbol{\Gamma}_i$ in Eq. (4.43) is the net externally applied torque on bead $\mathcal{B}_i$. The *generalized active torque* of the bead, $\boldsymbol{T}_i$, is hence given by the product of $\boldsymbol{\Gamma}_i$ and its partial angular velocity $\bar{\boldsymbol{\omega}}_i$:

$$\boldsymbol{T}_i \triangleq \sum \bar{\boldsymbol{\omega}}_i \circ \boldsymbol{\Gamma}_i. \tag{4.48}$$

Invoking Eq. (4.11), the above may be rewritten as:

$$\boldsymbol{T}_i = \bar{\boldsymbol{\omega}}_i^T \boldsymbol{\Gamma}_i = 2\boldsymbol{G}_i^T \boldsymbol{\Gamma}_i. \tag{4.49}$$

The generalized inertia force at the i^{th} bead is calculated using the definition:

$$\boldsymbol{F}_i^* \triangleq \sum \bar{\boldsymbol{v}}_i \circ \boldsymbol{f}_i^*, \tag{4.50}$$

where $\boldsymbol{f}_i^*$ is the inertia force calculated from:

$$\boldsymbol{f}_i^* = -m_b \boldsymbol{a}_i = -m_b \ddot{\boldsymbol{r}}_i. \tag{4.51}$$

Substitution of Eq. (4.51) into Eq. (4.50) then yields:

$$\boldsymbol{F}_i^* = \bar{\boldsymbol{v}}_i^T (-m_b \dot{\boldsymbol{r}}_i) = -m_b \boldsymbol{I}_3 \ddot{\boldsymbol{r}}_i. \tag{4.52}$$

The rate of change of angular momentum of the i^{th} bead, expressed in the inertial reference frame coordinates, is:

$$\dot{\boldsymbol{H}}_i = \boldsymbol{J}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{J}_i \boldsymbol{\omega}_i, \tag{4.53}$$

where $\boldsymbol{J}_i$ and $\boldsymbol{\omega}_i$ are the moments of inertia tensor and the angular velocity vector of the bead, both expressed in the inertial coordinates. Since the components of $\boldsymbol{J}_i$ in the inertial frame change values as the bead rotates in space, their direct use in the problem formulation could result in unnecessary complexity. Therefore, it is desirable in this case to express $\boldsymbol{J}_i$ in terms of its constant body-fixed components $\boldsymbol{J}_i^B$; *i.e.*,

$$\boldsymbol{J}_i = \boldsymbol{R}_i \boldsymbol{J}_i^B \boldsymbol{R}_i^T, \tag{4.54}$$

where $\boldsymbol{R}_i$ is the rotation matrix relating the inertial frame to the body-fixed frame, and $\boldsymbol{J}_i^B$ is the inertia tensor of the bead expressed in the body-fixed coordinates. If the bead is approximated by a sphere, $\boldsymbol{J}_i^B$ could be written as:

$$\boldsymbol{J}^B = J_o \boldsymbol{I}_3, \tag{4.55}$$

where $J_o = \frac{2}{5} m r_i^2$ is the polar moment of inertia of the bead. Hence, it is straightforward to show that:

$$\boldsymbol{J}_i = \boldsymbol{R}_i \boldsymbol{J}_i^B \boldsymbol{R}_i^T = \boldsymbol{R}_i J_o \boldsymbol{I}_3 \boldsymbol{R}^T = J_o \boldsymbol{I}_3 = \boldsymbol{J}_i^B, \tag{4.56}$$

which confirms the expectation that the inertia tensor of a sphere remains constant regardless of the frame of reference.

The angular velocity of the bead can be expressed in terms of its quaternion matrix $\boldsymbol{G}_i$ and the time derivative of the quaternion, $\dot{\boldsymbol{q}}_i$, as:

$$\boldsymbol{\omega}_i = 2\boldsymbol{G}_i\dot{\boldsymbol{q}}_i, \tag{4.57}$$

$$\dot{\boldsymbol{\omega}}_i = 2(\boldsymbol{G}_i\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{G}}_i\dot{\boldsymbol{q}}_i). \tag{4.58}$$

For unit quaternions, the product $\dot{\boldsymbol{G}}\dot{\boldsymbol{q}}$ equals to 0, resulting in:

$$\dot{\boldsymbol{\omega}}_i = 2\boldsymbol{G}_i\ddot{\boldsymbol{q}}_i \tag{4.59}$$

In addition, the cross product term $\boldsymbol{\omega}_i \times \boldsymbol{J}_i\boldsymbol{\omega}_i$ in Eq. (4.53) could be alternatively written as:

$$\boldsymbol{\omega}_i \times \boldsymbol{J}_i\boldsymbol{\omega}_i = [\boldsymbol{\Omega}]_\times \boldsymbol{J}_i\boldsymbol{\omega}_i, \tag{4.60}$$

where $[\boldsymbol{\Omega}_i]_\times$ is a skew-symmetric matrix in $\boldsymbol{\omega}_i$ defined by:

$$[\boldsymbol{\Omega}_i]_\times = \begin{bmatrix} 0 & -\omega_{iz} & \omega_{iy} \\ \omega_{iz} & 0 & -\omega_{ix} \\ -\omega_{iy} & \omega_{ix} & 0 \end{bmatrix} = 2\dot{\boldsymbol{G}}_i\boldsymbol{G}_i^T \tag{4.61}$$

Substitution of the expressions in Eq. (4.56) and Eq. (4.59) into Eq. (4.53) and using the cross-product identity from Eq. (4.61) yields:

$$\dot{\boldsymbol{H}}_i = 2J_o\boldsymbol{G}_i\ddot{\boldsymbol{q}}_i + 4J_o\dot{\boldsymbol{G}}\dot{\boldsymbol{q}}_i - \ 4J_o\dot{\boldsymbol{G}}_i\boldsymbol{q}_i\boldsymbol{q}_i^T\dot{\boldsymbol{q}}_i \tag{4.62}$$

Since the $\dot{\boldsymbol{G}}_i\dot{\boldsymbol{q}}_i$ and $\boldsymbol{q}_i^T\dot{\boldsymbol{q}}_i$ terms vanish for unit quaternions, the above further reduces to:

$$\dot{\boldsymbol{H}}_i = 2J_o\boldsymbol{G}_i\ddot{\boldsymbol{q}}. \tag{4.63}$$

The *generalized inertia torque* may be calculated from the rate of change of angular momentum and the angular partial velocity matrix as:

$$\boldsymbol{T}_i^* \triangleq \sum \bar{\boldsymbol{\omega}}_i \circ (-\dot{\boldsymbol{H}}_i) = -\bar{\boldsymbol{\omega}}_i^T\dot{\boldsymbol{H}}_i. \tag{4.64}$$

Substitution of Eq. (4.14) and Eq. (4.63) into Eq. (4.64) and further simplification furnishes:

$$\boldsymbol{T}_i^* = -(4J_o\boldsymbol{I}_4\ddot{\boldsymbol{q}}_i - 4J_o\boldsymbol{q}_i\boldsymbol{q}_i^T\ddot{\boldsymbol{q}}_i). \tag{4.65}$$

In addition, differentiating twice the unit quaternion norm in Eq. (4.4) with respect to time yields the identity:

$$\boldsymbol{q}^T\ddot{\boldsymbol{q}} = -\|\dot{\boldsymbol{q}}\|^2. \tag{4.66}$$

The above is substituted into Eq. (4.65) to derive the following expression for the *generalized inertia torque* of the bead:

$$\boldsymbol{T}_i^* = -4J_o\boldsymbol{I}_4\ddot{\boldsymbol{q}}_i - 4J_o\|\dot{\boldsymbol{q}}_i\|^2\boldsymbol{q}. \tag{4.67}$$

4.2.5 Unconstrained Equations of Motion using Kane's Method

Kane's dynamical equations is a streamlined generalized approach for deriving equations of motion. This method has roots in Lagrangian Dynamics and D'Alembert's principle [339–343]. One of the advantages of this method over the Newton-Euler method is the absence of non-contributing forces in the problem formulation, which significantly simplifies the process. In addition, although this method is developed based on Lagrangian dynamics, it does not require the derivation of the total energy or the Lagrangian of the system. The equations are obtained by taking the sum of the generalized active forces, $\boldsymbol{F}$, and generalized inertia forces, $\boldsymbol{F}^*$, defined in the previous section (see Eq. (4.46) and (4.50). The resulting expressions in Equation 4.68, also known as *Kane's dynamical equations*, furnish the equations of motion; *i.e.*,

$$\boldsymbol{F} + \boldsymbol{F}^* = \boldsymbol{0} \tag{4.68}$$

Similarly, the generalized active torques, $\boldsymbol{T}$, and generalized inertia torques, $\boldsymbol{T}^*$, are summed to obtain the rotational equations of motion of the system, *i.e.*,

$$\boldsymbol{T} + \boldsymbol{T}^* = \boldsymbol{0} \tag{4.69}$$

The complete unconstrained equations of motion for the i^{th} bead are then obtained

by combining the above two equations; *i.e.*,

$$\boldsymbol{F}_i + \boldsymbol{F}_i^* = \boldsymbol{0} \tag{4.70}$$

$$\boldsymbol{T}_i + \boldsymbol{T}_i^* = \boldsymbol{0} \tag{4.71}$$

Substitution of the expressions in Eq. (4.47) and (4.52) into Eq. (4.70) for $\boldsymbol{F}_i$ and $\boldsymbol{F}_i^*$, respectively, yields the transnational equation of motion of the bead:

$$m_i \boldsymbol{I}_3 \ddot{\boldsymbol{r}}_i = \boldsymbol{I}_3 \boldsymbol{R}_i. \tag{4.72}$$

Equation (4.72) is the standard form of equations of motion (*i.e.*, $\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{F}$), where $m_i \boldsymbol{I}_3$ is the mass matrix of the bead, and $\boldsymbol{I}_3 \boldsymbol{R}_i$ is the *impressed force*. Similarly, the unconstrained rotational equations of motion of the bead are determined by adding the generalized active torques and generalized inertial torques (see Eq. (4.49) and Eq. (4.67)):

$$\begin{gathered} \boldsymbol{T}_i^* + \boldsymbol{T}_i = \boldsymbol{0}, \\ -4J_o \boldsymbol{I}_4 \ddot{\boldsymbol{q}}_i - 4J_o \|\dot{\boldsymbol{q}}_i\|^2 \boldsymbol{q}_i + \boldsymbol{G}_i \boldsymbol{E}_i^T \boldsymbol{\Gamma}_i^B = 0_{4\times 1}. \end{gathered} \tag{4.73}$$

where $0_{4\times 1}$ is a 4-by-1 zero matrix.

A simple rearrangement of Eq. (4.73) furnishes the standard form of the rotational equations of motion; *i.e.*,

$$4J_o \boldsymbol{I}_4 \ddot{\boldsymbol{q}}_i = -4J_o \|\dot{\boldsymbol{q}}_i\|^2 \boldsymbol{q}_i + \boldsymbol{G}_i \boldsymbol{E}_i^T \boldsymbol{\Gamma}_i^B, \tag{4.74}$$

where the coefficient of $\ddot{\boldsymbol{q}}_i$ in the above equation is the effective moment of inertia matrix of the bead in question and is denoted by $\boldsymbol{J}_i^{'}$:

$$\boldsymbol{J}_i^{'} = 4J_o \boldsymbol{I}_4, \tag{4.75}$$

where $\boldsymbol{I}_4$ is a 4-by-4 identity matrix.

The combined term on the right hand side of Eq. (4.73) represents the *impressed torque*, $\boldsymbol{Q}'$, where

$$\boldsymbol{Q}' = -4J_o \|\dot{\boldsymbol{q}}_i\|^2 \boldsymbol{q}_+ \boldsymbol{G}_i \boldsymbol{E}_i^T \boldsymbol{\Gamma}_i^B. \tag{4.76}$$

The unconstrained transnational and rotational equation of motions in Eq. (4.72) and Eq. (4.74) may be recast into a matrix form as:

$$\begin{bmatrix} m_b \boldsymbol{I}_3 & 0 \\ 0 & 4J_o \boldsymbol{I}_4 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{r}}_i \\ \ddot{\boldsymbol{q}}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 \boldsymbol{R}_i \\ -4J_o \|\dot{\boldsymbol{q}}_i\|^2 \boldsymbol{q}_i + \boldsymbol{G}_i \boldsymbol{E}_i^T \boldsymbol{\Gamma}_i^B \end{bmatrix}, \tag{4.77}$$

or equivalently,

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i = \boldsymbol{Q}_i \tag{4.78}$$

where $\boldsymbol{M}_i$, the effective mass matrix of the bead, and $\boldsymbol{Q}_i$, the combined vector of impressed forces and torques, are given by:

$$\boldsymbol{M}_i = \begin{bmatrix} m_i \boldsymbol{I}_3 & 0 \\ 0 & 4J_o \boldsymbol{I}_4 \end{bmatrix}, \qquad \boldsymbol{Q}_i = \begin{bmatrix} \boldsymbol{I}_3 \boldsymbol{R}_i \\ -4J_o \|\dot{\boldsymbol{q}}_i\|^2 \boldsymbol{q}_i + \boldsymbol{G}_i \boldsymbol{E}_i^T \boldsymbol{\Gamma}_i^B \end{bmatrix}. \tag{4.79}$$

The unconstrained equation of motion in Eq. (4.78) is only for a single local bead, namely $\mathcal{B}_i$. The equations of motion for the overall system are formulated by systematically assembling the local equations of motion into a global matrix (strictly diagonal):

$$\begin{bmatrix} \boldsymbol{M}_1 & & & & \\ & \ddots & & & \\ & & \boldsymbol{M}_i & & \\ & & & \ddots & \\ & & & & \boldsymbol{M}_N \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_1 \\ \vdots \\ \ddot{\boldsymbol{u}}_i \\ \vdots \\ \ddot{\boldsymbol{u}}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_1 \\ \vdots \\ \boldsymbol{Q}_i \\ \vdots \\ \boldsymbol{Q}_N \end{bmatrix}. \tag{4.80}$$

The system's global mass matrix and impressed global forces and torques vectors, respectively, in Eq. (4.80) are denoted by $\boldsymbol{M}$ and $\boldsymbol{Q}$, leading to a compact expression:

$$\boldsymbol{M} \ddot{\boldsymbol{u}} = \boldsymbol{Q}. \tag{4.81}$$

4.2.6 Constrained Equations of Motion using Udwadia-Kalaba Method

The unconstrained system of equations in Eq. (4.81) presume that the beads in the system are free to translate and rotate in any direction. However, the motions of the beads are subject to restrictions due to the viscoelastic joints and restrictions arising from beads sticking to the collector plate. Such kinematic restrictions are imposed on the unconstrained system in the form of constraint forces using the Udwadia-Kalaba method [338]. Unlike other techniques which employ eigenvalues [339, 379],

this method does not require solving for extra intermediate variables (eigenvalues), lessening the complexity of the problem and reducing computational requirements.

The constraint equations must be identified and differentiated with respect to time once or twice, depending on the nature of the constraints. As shown in the upcoming sections, since all constraints in the system are exclusively holonomic, they are differentiated twice. These constraints are normally written in the vector form:

$$\boldsymbol{\varphi}(\boldsymbol{u}, t) = \mathbf{0}. \tag{4.82}$$

Differentiating the constraints twice with respect to time leads to the following:

$$\frac{d^2}{dt^2}\boldsymbol{\varphi}(\boldsymbol{u}, t) = \ddot{\boldsymbol{\varphi}}(\ddot{\boldsymbol{u}}, \dot{\boldsymbol{u}}, \boldsymbol{u}, t) = \mathbf{0}. \tag{4.83}$$

The resulting expression in Eq. (4.83) may then be rearranged into a matrix form by denoting the coefficient of the acceleration vector $\ddot{\boldsymbol{u}}$ by matrix $\boldsymbol{A}$, and collecting the remaining terms into vector $\boldsymbol{b}$. *i.e.*,

$$\boldsymbol{A}(\dot{\boldsymbol{u}}, \boldsymbol{u}, t)\ddot{\boldsymbol{u}} = \boldsymbol{b}(\dot{\boldsymbol{u}}, \boldsymbol{u}, t). \tag{4.84}$$

Body Constraints - Norm of Unit Quaternion

Each bead in the system is assumed to have a unique orientation and hence has a unique set of quaternions to describe it. Since the norm of a unit quaternion is 1, each bead contributes one constraint equation. Precisely, the norm of the i^{th} bead's unit quaternion is:

$$\|\boldsymbol{q}_i\|^2 = 1. \tag{4.85}$$

The above leads to the constraint equation:

$$\varphi_{i1}(\boldsymbol{q}_i, t): \quad \|\boldsymbol{q}_i\|^2 - 1 = 0. \tag{4.86}$$

It is noted that Eq. (4.86) is purely geometric, hence is a holonomic constraint. Differentiating this constraint equation twice with respect to time yields:

$$\frac{\partial^2 \varphi_{i1}}{\partial t^2} = \|\dot{\boldsymbol{q}}_i\|^2 + \boldsymbol{q}_i^T \ddot{\boldsymbol{q}}_i = 0. \tag{4.87}$$

Equation (4.87) may be rearranged to get:

$$\boldsymbol{q}_i^T \ddot{\boldsymbol{q}}_i = -\|\dot{\boldsymbol{q}}_i\|^2, \tag{4.88}$$

which fits into the $\boldsymbol{A}_{qi}\ddot{\boldsymbol{q}}_i = \boldsymbol{b}_{qi}$ matrix form, where

$$\boldsymbol{A}_{qi} = \boldsymbol{q}_i^T, \quad \boldsymbol{b}_{qi} = -\|\dot{\boldsymbol{q}}_i\|^2. \tag{4.89}$$

Similar constraints contributed by all other beads are assembled into a single matrix to obtain the *global body constraint equation matrix*; *i.e.*,

$$\begin{bmatrix} [0_{1\times 3} \mid \boldsymbol{A}_{q1}] & & & & \\ & \ddots & & & \\ & & [0_{1\times 3} \mid \boldsymbol{A}_{qi}] & & \\ & & & \ddots & \\ & & & & [0_{1\times 3} \mid \boldsymbol{A}_{qN}] \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_1 \\ \vdots \\ \ddot{\boldsymbol{u}}_i \\ \vdots \\ \ddot{\boldsymbol{u}}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{q1} \\ \vdots \\ \boldsymbol{b}_{qi} \\ \vdots \\ \boldsymbol{b}_{qN} \end{bmatrix}. \tag{4.90}$$

For brevity, the coefficient matrix and the vector on the right-hand side of Eq. (4.90) are denoted by $\boldsymbol{A}_B$ and $\boldsymbol{b}_B$, respectively, to obtain the following compact form:

$$\boldsymbol{A}_B\ddot{\boldsymbol{u}} = \boldsymbol{b}_B. \tag{4.91}$$

Viscoelastic Joint Constraints

The three mutually perpendicular unit vectors $\hat{\boldsymbol{i}}_o$, $\hat{\boldsymbol{j}}_o$, and $\hat{\boldsymbol{k}}_o$ in Fig. 4.7 may be expressed by their components as:

$$\hat{\boldsymbol{i}}_o = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{\boldsymbol{j}}_o = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \hat{\boldsymbol{k}}_o = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4.92}$$

Equivalently, they can be collected into a single matrix form:

$$\begin{bmatrix} \hat{\boldsymbol{i}}_o & \hat{\boldsymbol{j}}_o & \hat{\boldsymbol{k}}_o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \boldsymbol{I}_3. \tag{4.93}$$

If these unit vectors are rotated using the rotation matrix of the i^{th} bead, they become parallel to the bead's body-fixed coordinate system. The rotated unit vectors, denoted by $\hat{\boldsymbol{i}}'_o$, $\hat{\boldsymbol{j}}'_o$, and $\hat{\boldsymbol{j}}'_o$, are determined from:

$$\begin{bmatrix} \hat{\boldsymbol{i}}'_o & \hat{\boldsymbol{j}}'_o & \hat{\boldsymbol{k}}'_o \end{bmatrix} = \boldsymbol{R}_i \begin{bmatrix} \hat{\boldsymbol{i}}_o & \hat{\boldsymbol{j}}_o & \hat{\boldsymbol{k}}_o \end{bmatrix} = \boldsymbol{R}_i, \tag{4.94}$$

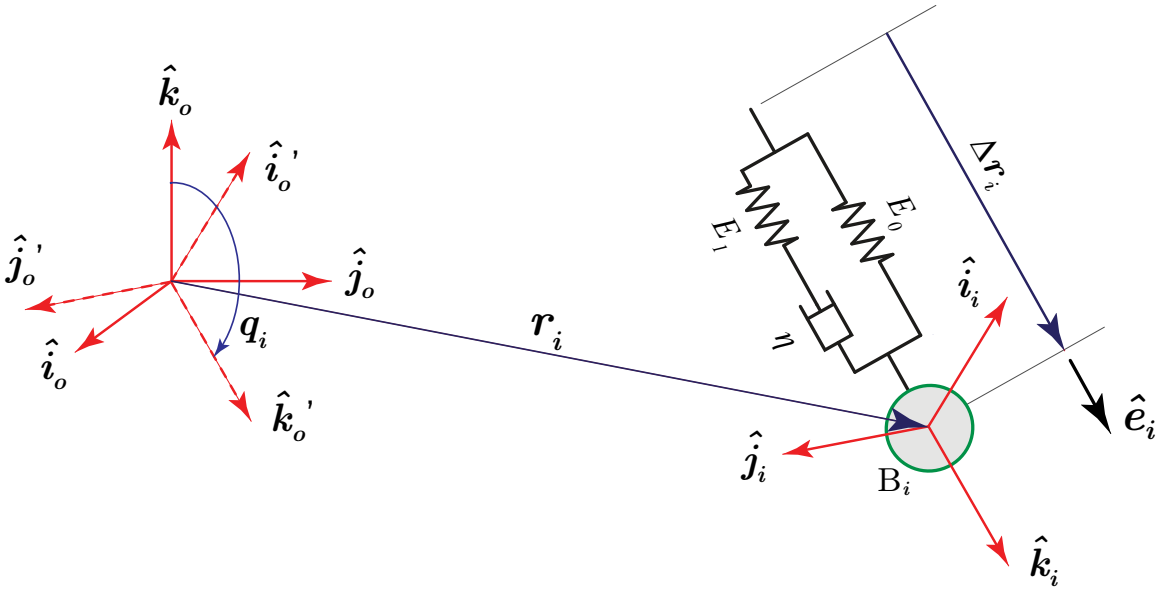


Figure 4.7: Viscoelastic joint constraint configuration

where $\boldsymbol{R}_i$ is the rotation matrix of the i^{th} bead,

$$\boldsymbol{R}_i(\boldsymbol{q}_i) = \begin{bmatrix} q_{i0}^2 + q_{i1}^2 - q_{i2}^2 - q_{i3}^2 & 2q_{i1}q_{i2} - 2q_{i0}q_{i3} & 2q_{i0}q_{i2} + 2q_{i1}q_{i3} \\ 2q_{i0}q_{i3} + 2q_{i1}q_{i2} & q_{i0}^2 - q_{i1}^2 + q_{i2}^2 - q_{i3}^2 & 2q_{i2}q_{i3} - 2q_{i0}q_{i1} \\ 2q_{i1}q_{i3} - 2q_{i0}q_{i2} & 2q_{i0}q_{i1} + 2q_{i2}q_{i3} & q_{i0}^2 - q_{i1}^2 - q_{i2}^2 + q_{i3}^2, \end{bmatrix} \tag{4.95}$$

The rotated unit vectors $\hat{\boldsymbol{i}}_o'$ and $\hat{\boldsymbol{j}}_o'$, in this case, are expressed in terms of components of the rotation matrix:

$$\hat{\boldsymbol{i}}_o' = \begin{bmatrix} q_{i0}^2 + q_{i1}^2 - q_{i2}^2 - q_{i3}^2 \\ 2q_{i0}q_{i3} + 2q_{i1}q_{i2} \\ 2q_{i1}q_{i3} - 2q_{i0}q_{i2} \end{bmatrix}, \quad \hat{\boldsymbol{j}}_o' = \begin{bmatrix} 2q_{i1}q_{i2} - 2q_{i0}q_{i3} \\ q_{i0}^2 - q_{i1}^2 + q_{i2}^2 - q_{i3}^2 \\ 2q_{i0}q_{i1} + 2q_{i2}q_{i3} \end{bmatrix}. \tag{4.96}$$

The constraint equations for the i^{th} viscoelastic joint are derived from the observation that the rotated unit vectors ($\hat{\boldsymbol{i}}_o'$ and $\hat{\boldsymbol{j}}_o'$) are parallel to their respective unit vectors ($\hat{\boldsymbol{i}}_i$ and $\hat{\boldsymbol{j}}_i$) in the body-fixed coordinate system. As shown in Fig. 4.7, both $\hat{\boldsymbol{i}}_o'$ and $\hat{\boldsymbol{j}}_o'$ are also perpendicular to the displacement vector $\Delta\boldsymbol{r}_i$, hence their respective dot products with $\Delta\boldsymbol{r}_i$ vanish; *i.e.*,

$$\hat{\boldsymbol{i}}_o' \cdot \Delta\boldsymbol{r}_i = 0, \tag{4.97}$$

$$\hat{\boldsymbol{j}}_o' \cdot \Delta\boldsymbol{r}_i = 0. \tag{4.98}$$

The rotated vector $\hat{\boldsymbol{k}}_o'$ is parallel to and pointing in the same direction as $\Delta\boldsymbol{r}_i$. Since the norm of $\hat{\boldsymbol{k}}_o'$ is 1, and the angle between these two vectors is zero, the dot

product between $\hat{\boldsymbol{k}}_o^{'}$ and $\Delta \boldsymbol{r}_i$ is simply the norm of $\Delta \boldsymbol{r}_i$. *i.e.*,

$$\hat{\boldsymbol{k}}_o^{'} \cdot \Delta \boldsymbol{r}_i = \|\Delta \boldsymbol{r}_i\|. \tag{4.99}$$

Equation (4.97) – (4.99) constitute the viscoelastic joint constraint equations. However, since a right-hand coordinate system could be established from nobly two axes, two out of the three constraint equations are adequate. Namely, Eq. (4.97) and (4.98) are adapted here as constraint equations. In fact, using all three would result in an overly constrained system of equations and thus should be avoided.

Using the $\hat{\boldsymbol{i}}_o^{'}$ and $\hat{\boldsymbol{j}}_o^{'}$ expressions from Eq. (4.96) in Eq. (4.97) – (4.98), and invoking the definition of $\Delta \boldsymbol{r}_i$ from Eq. (4.25), the viscoelastic joint constraints of the i^{th} link become:

$$\varphi_{i2}(\boldsymbol{r}_{i-1}, \boldsymbol{r}_i, \boldsymbol{q}_i, t): \quad \begin{bmatrix} q_{i0}^2 + q_{i1}^2 - q_{i2}^2 - q_{i3}^2 \\ 2q_{i0}q_{i3} + 2q_{i1}q_{i2} \\ 2q_{i1}q_{i3} - 2q_{i0}q_{i2} \end{bmatrix} \cdot \begin{bmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \\ z_i - z_{i-1} \end{bmatrix} = 0, \tag{4.100}$$

$$\varphi_{i3}(\boldsymbol{r}_{i-1}, \boldsymbol{r}_i, \boldsymbol{q}_i, t): \quad \begin{bmatrix} 2q_{i1}q_{i2} - 2q_{i0}q_{i3} \\ q_{i0}^2 - q_{i1}^2 + q_{i2}^2 - q_{i3}^2 \\ 2q_{i0}q_{i1} + 2q_{i2}q_{i3} \end{bmatrix} \cdot \begin{bmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \\ z_i - z_{i-1} \end{bmatrix} = 0. \tag{4.101}$$

These constraints are holonomic and must be converted into the $\boldsymbol{A}\ddot{\boldsymbol{u}} = \boldsymbol{b}$ form by differentiating twice with respect to time. Here we present only the final results for the sake of conciseness.

$$\begin{bmatrix} -\boldsymbol{R}_i^T & 0_{3\times 4} & \boldsymbol{R}_i^T & 2\boldsymbol{E}_i[\Delta \boldsymbol{r}_i]_\times \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_{i-1} \\ \ddot{\boldsymbol{u}}_i \end{bmatrix} = -\left[4\boldsymbol{E}_i \dot{\boldsymbol{G}}_i^T \Delta \dot{\boldsymbol{r}}_i + 2\dot{\boldsymbol{E}}_i \dot{\boldsymbol{G}}_i^T \Delta \boldsymbol{r}_i \right], \tag{4.102}$$

where $\boldsymbol{E}_i$ is a matrix linear in the quaternion elements. It is defined as:

$$\boldsymbol{E}(\boldsymbol{q}) \triangleq \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \tag{4.103}$$

$[\Delta \boldsymbol{r}_i]_\times$ is a 4-by-4 skew-symmetric matrix in terms of the position generalized coordinates:

$$[\Delta \boldsymbol{r}_i]_\times = \begin{bmatrix} 0 & x_{i-1} - x_i & y_{i-1} - y_i & z_{i-1} - z_i \\ x_i - x_{i-1} & 0 & z_{i-1} - z_i & y_i - y_{i-1} \\ y_i - y_{i-1} & z_i - z_{i-1} & 0 & x_{i-1} - x_i \\ z_i - z_{i-1} & y_{i-1} - y_i & x_i - x_{i-1} & 0 \end{bmatrix}. \tag{4.104}$$

Equation (4.102) represents three viscoelastic joint constraint equations in the form of:

$$\boldsymbol{A}_{Ji}\begin{bmatrix}\ddot{\boldsymbol{u}}_{i-1}\\ \ddot{\boldsymbol{u}}_i\end{bmatrix} = \boldsymbol{b}_{Ji}, \tag{4.105}$$

where

$$\boldsymbol{A}_{Ji} = \begin{bmatrix}-\boldsymbol{R}_i^T & 0_{3\times4} & \boldsymbol{R}_i^T & 2\boldsymbol{E}_i[\Delta\boldsymbol{r}_i]_\times\end{bmatrix}, \tag{4.106}$$

$$\boldsymbol{b}_{Ji} = -\left[4\boldsymbol{E}_i\dot{\boldsymbol{G}}_i^T(\Delta\dot{\boldsymbol{r}}_i) + 2\dot{\boldsymbol{E}}_i\dot{\boldsymbol{G}}_i^T(\Delta\boldsymbol{r}_i)\right]. \tag{4.107}$$

However, as stated above, since only two of the three constraints are sufficient, the last rows of $\boldsymbol{A}_{Ji}$ and $\boldsymbol{b}_{Ji}$ are omitted from the constraint equations. $\boldsymbol{A}_{Ji}$ is further partitioned into $\boldsymbol{A}_{Ji-}$ and $\boldsymbol{A}_{Ji+}$ to make assembling the global joint constraint matrix an easier task; *i.e.*,

$$\boldsymbol{A}_{Ji} = [\boldsymbol{A}_{Ji-} \mid \boldsymbol{A}_{Ji+}], \tag{4.108}$$

where

$$\boldsymbol{A}_{Ji-} = \begin{bmatrix}-\boldsymbol{R}_i^T & 0_{3\times4}\end{bmatrix}, \qquad \boldsymbol{A}_{Ji+} = \begin{bmatrix}\boldsymbol{R}_i^T & 2\boldsymbol{E}_i[\Delta\boldsymbol{r}_i]_\times\end{bmatrix}. \tag{4.109}$$

When the value of i is 1, $\boldsymbol{A}_{Ji-} = \boldsymbol{A}_{J0}$. The index 0 represents the nozzle (or spinneret) to which the first link is connected. As such, $\ddot{\boldsymbol{u}}_0$ represents the linear and angular accelerations of the nozzle. While the orientation of the nozzle can be set to any desired direction when setting up the MES apparatus, it is assumed to be a constant parameter once the simulation commences. Hence is not a time-dependent variable (*i.e.*, the nozzle's angular velocity and angular acceleration are zero.) On the other hand, if the nozzle travels in a predefined path, as would be the case in direct-writing or 3D printing applications, it would have its own time-dependent position, velocity, and acceleration variables. Accordingly, the constraint expressions for the first link are modified to reflect these observations.

$$\begin{bmatrix}-\boldsymbol{R}_1^T & 0_{3\times4} & \boldsymbol{R}_1^T & 2\boldsymbol{E}_1[\Delta\boldsymbol{r}_1]_\times\end{bmatrix}\begin{bmatrix}\ddot{\boldsymbol{u}}_0\\ \ddot{\boldsymbol{u}}_1\end{bmatrix} = \boldsymbol{b}_{J1}, \tag{4.110}$$

which may be rearranged to furnish the following:

$$\begin{bmatrix}\boldsymbol{R}_1^T & 2\boldsymbol{E}_1[\Delta\boldsymbol{r}_1]_\times\end{bmatrix}\ddot{\boldsymbol{u}}_1 = \boldsymbol{b}_{J1} - \begin{bmatrix}-\boldsymbol{R}_1^T & 0_{3\times4}\end{bmatrix}\ddot{\boldsymbol{u}}_0 = \boldsymbol{b}_{J1} + \boldsymbol{R}_1^T\ddot{\boldsymbol{r}}_0. \tag{4.111}$$

The joint constraint equations developed so far are only for a single generic joint, namely joint $\mathcal{J}_i$. Constraints representing the other joints may be systematically generated in a similar manner. The results are assembled into a single matrix to create a *system of joint constraint equations*:

$$\begin{bmatrix} \boldsymbol{A}_{J1+} & & & & & & \\ \boldsymbol{A}_{J2-} & \boldsymbol{A}_{J2+} & & & & & \\ & \boldsymbol{A}_{J3-} & \boldsymbol{A}_{J3+} & & & & \\ & & \boldsymbol{A}_{J4-} & \boldsymbol{A}_{J4+} & & & \\ & & & \ddots & & & \\ & & & & & \boldsymbol{A}_{JN-} & \boldsymbol{A}_{JN+} \end{bmatrix} \ddot{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{b}_{J1} + \boldsymbol{R}_1^T \ddot{\boldsymbol{r}}_{noz} \\ \boldsymbol{b}_{J2} \\ \boldsymbol{b}_{J3} \\ \boldsymbol{b}_{J4} \\ \vdots \\ \boldsymbol{b}_{JN} \end{bmatrix}. \tag{4.112}$$

Equivalently, this may be written in a compact form as:

$$\boldsymbol{A}_J \ddot{\boldsymbol{u}} = \boldsymbol{b}_J, \tag{4.113}$$

where $\boldsymbol{A}_J$ and $\boldsymbol{b}_J$ are the coefficient matrix and the right-hand side term in Eq. (4.112), respectively.

Table-sticking (adhering) Constraints

Once the free end of the melt-electrospun fiber touches down and sticks to the collector plate, it anchors the rest of the fiber to the contact point, essentially resisting free fiber spinning. The beads in the stuck fiber segment are assumed to have no translational movement relative to the plate. This sticking phenomenon affects the dynamics of the remainder of the active fiber still in the air. Unlike the previous two types of constraints, this constraint is selectively applied to the transnational motions of the last few beads during an iteration. The collector distance is used as an event trigger criterion in the simulation to identify beads that have reached the surface of the collector plate. If the k^{th} bead has touched down on the collector plate during the simulation, the position vector of the bead is assumed to be locked to the Cartesian coordinates of the contact point (static variables). If the position vector of this bead and the coordinates of its contact point are denoted, respectively, by $\boldsymbol{r}_k = [x_k \;\; y_k \;\; z_k]^T$ and $\boldsymbol{r}_k^* = [x_k^* \;\; y_k^* \;\; z_k^*]^T$, the sticking constraint may be written as:

$$\boldsymbol{\varphi}_k(\boldsymbol{r}_k, t): \quad \boldsymbol{r}_k - \boldsymbol{r}_k^* = \boldsymbol{0}. \tag{4.114}$$

Equation (4.114) yields three constraint equations per stuck bead, in agreement with the observation that translations along the global x, y, and z directions are restricted.

These constraints are purely holonomic and hence are differentiated twice with respect to time; *i.e.*,

$$\frac{\partial^2 \boldsymbol{\varphi}_k}{\partial t^2} = \frac{\partial^2}{\partial t^2}\left(\boldsymbol{r}_k - \boldsymbol{r}_k^*\right) = \ddot{\boldsymbol{r}}_k = \boldsymbol{I}_3 \ddot{\boldsymbol{r}}_k = [0 \;\; 0 \;\; 0]^T. \tag{4.115}$$

Equation (4.115) is in the $\boldsymbol{A}_k \ddot{\boldsymbol{u}}_k = \boldsymbol{b}_k$ format, where:

$$\boldsymbol{A}_k = [I_3], \quad \boldsymbol{b}_k = [0 \;\; 0 \;\; 0]^T. \tag{4.116}$$

4.2.7 Constrained Equation of Motions

The constraint equations derived so far in Eq. (4.91), (4.113), and (4.116) are collected into a single matrix equation that represents the combined constraints in the system.

$$\begin{bmatrix} \boldsymbol{A}_B \\ \boldsymbol{A}_J \\ \boldsymbol{A}_k \end{bmatrix} \ddot{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{b}_B \\ \boldsymbol{b}_J \\ \boldsymbol{b}_k \end{bmatrix}. \tag{4.117}$$

Alternatively, rewritten in an equivalent compact notation:

$$\boldsymbol{A}\ddot{\boldsymbol{u}} = \boldsymbol{b}, \tag{4.118}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_B \\ \boldsymbol{A}_J \\ \boldsymbol{A}_k \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_B \\ \boldsymbol{b}_J \\ \boldsymbol{b}_k \end{bmatrix}. \tag{4.119}$$

The Udwadia-Kalaba method is employed here to convert the constraints in Eq. (4.118) into *constraint forces*. This method works by calculating a new set of forces and torques, denoted by $\boldsymbol{Q}^c$, that is required to enforce the constraints on the system. $\boldsymbol{Q}^c$ is calculated from the parameters of the unconstrained equations of motion (Eq. (4.81)) and the system's constraint equations (Eq. (4.118)) using the formula:

$$\boldsymbol{Q}^c = \boldsymbol{M}^{1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^{+}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}), \tag{4.120}$$

where the operator $(\,)^+$ is the Moore-Penrose inverse [344, 345].

Once the constraint force $\boldsymbol{Q}^c$ is known, it is added to $\boldsymbol{Q}$ in Eq. (4.81) to obtain the constrained equations of motion of the system; *i.e.*,

$$\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{Q} + \boldsymbol{Q}^c, \tag{4.121}$$

$$\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{Q} + \boldsymbol{M}^{1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^{+}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}). \tag{4.122}$$

The time histories of the beads (*i.e.*, their positions, orientations, transnational and angular velocities) are determined by numerically integrating Eq. (4.122). Proper and compatible initial conditions are required for this.

4.3 Model Implementation and Discussion

To numerically integrate and solve the equations of motion in Eq. (4.122) using commercially available or open-source multi-paradigm programming software, we form a complete set of algebraic differential equations (DAE) made up of Eq. (4.122) and other auxiliary variables; *i.e.*,

$$\begin{bmatrix} \boldsymbol{I}_{7N} & & \\ & \boldsymbol{M} & \\ & & \boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \ddot{\boldsymbol{u}} \\ \dot{\boldsymbol{\sigma}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{u}} \\ \boldsymbol{Q} + \boldsymbol{M}^{1/2}(\boldsymbol{A}\boldsymbol{M}^{-1/2})^{+}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{a}) \\ \dot{\boldsymbol{\sigma}}(\sigma, \boldsymbol{u}, \ddot{\boldsymbol{u}}, p, t) \end{bmatrix}, \tag{4.123}$$

where $\dot{\boldsymbol{\sigma}}(\sigma, \boldsymbol{u}, \ddot{\boldsymbol{u}}, p, t)$ is a vector function obtained by rearranging the governing equation in Eq. (4.22); *i.e.*,

$$\dot{\sigma}_i(\sigma_i, \boldsymbol{u}_i, \ddot{\boldsymbol{u}}_i, p, t) = \frac{E_1}{\eta}\left(-\sigma_i(t) + E_0\ \varepsilon_i(t) + \frac{\eta\ (E_0 + E_1)}{E_1}\ \dot{\varepsilon}_i(t)\right). \tag{4.124}$$

The DAE in Eq. (4.123) can be numerically integrated using suitable initial conditions for the dynamic variables:

$$\begin{bmatrix} \boldsymbol{u}^T & \dot{\boldsymbol{u}}^T & \boldsymbol{\sigma}^T \end{bmatrix}^T. \tag{4.125}$$

One of the objectives of the proposed work was to simulate MES with no limitation on the total duration and without the need to predefine the number of beads/links in the system. To this end, an algorithm is proposed (Fig. 4.8) to determine the model's size during each iteration dynamically. It continuously formulates and solves the constrained equations of motion in Eq. (4.122). Adaptive model size means there is no need to predefine the model's size ahead of the simulation, resulting in more efficient utilization of memory and computational resources. This capability is vital as it avoids allocating large memory to handle all scenarios, including the initial transient stages where only a few active beads play a meaningful role. The classical

approach for handling similar problems involves pre-allocating a large model size to handle the maximum number of beads the system could generate. This approach had two problems: 1) it is challenging to estimate the maximum model size, and 2) allocating "sufficiently" extra capacity could lead to underutilization. The algorithm proposed here addresses this by generating the correct model size at every iteration stage, leading to the optimum utilization of computational resources.

The beads in the system are categorized into three groups depending on their status during the simulation. The first set contains *active beads*, which are actively involved in the simulation. Upon touching down on the collector plate, active beads are considered stuck and unable to translate in any direction. Such beads constitute the second category known as *stuck beads*. Stuck beads are holonomically constrained to remain in the plane of the collector plate. They are actively involved in the numerical solution because of the boundary conditions they impose on the rest of the system. They lose their translational degrees of freedom because of the sticking but retain their rotational freedom to allow smooth fiber curvature formations. The last group of beads, named *frozen beads*, have already undergone the simulation process. They are buried under the stuck beads and do not influence the fiber dynamics. The coordinates of their sticking points (final resting places) on the collector plate have been stored for the record. Frozen beads are omitted from the formulation when the simulation runs in the subsequent iterations, dynamically adjusting the model's size and maintaining efficient memory use. The DAE in Eq. (4.123) was implemented in MATLAB®using the ode45 function (A). The Baumgarte numerical stabilization method [380–383] (with $\alpha = 20$ and $\beta = 20$) was also employed to stabilize the numerical solution around the constraints.

The model was first run to demonstrate simple directional MES configurations. For example, Fig. 4.9 and Fig. 4.10 show the simulation outputs for a vertical and a 45° spinning angle configurations, respectively. Additional vertical simulations were carried out to demonstrate the scenario in which the nozzle is set to travel along a prescribed path, as in most 3D printing operations. As an example, in Fig. 4.11, the nozzle is forced to follow a sinusoidal path defined by Equation (4.126 with extremely high translational speeds (up to $50\,\mathrm{m\,s^{-1}}$ in the x axis direction). This is intentionally

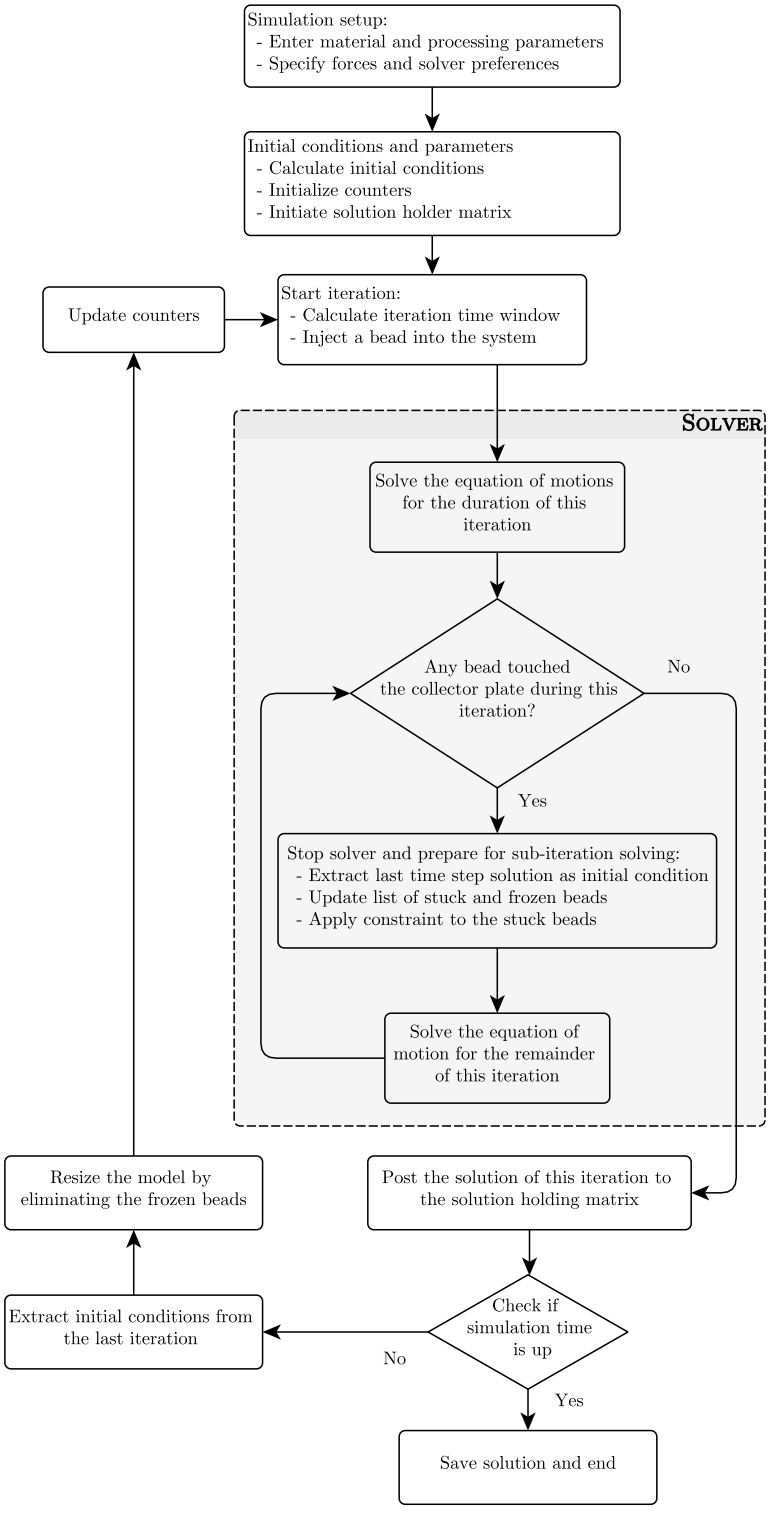


Figure 4.8: Flowchart showing programming implementation strategy

done to produce an exaggerated fiber whipping and visually bring to evidence the effects of the table sticking phenomena, which otherwise would be very difficult to show. It is worth noting, however, that the speeds of the nozzles or the collector tables in most 3D printers are much lower than this (as low as $180\,\mathrm{mm\,s^{-1}}$ [352, 384]).

$$x(t) = 50t\,\mathrm{m}, \quad y(t) = 0.025\,\sin(1.25t)\,\mathrm{m}, \quad z(t) = 0\,\mathrm{m}. \tag{4.126}$$

Next, the model was tested with three sets of vertical MES simulations designed to investigate the sensitivity of the morphological and topological output parameters to changes in process parameters – namely applied voltage, collector distance, and flow rate. During these simulations, measurements of the fiber diameter, collection diameter, maximum jet speed, and fiber elongation percentages were taken. In the first set of simulations, the collector distance and the flow rate were kept constant while the applied voltage was increased from 5 to 20 kV with increments of 5 kV (Table 4.1). In the second, the applied voltage and the flow rate were kept constant while the collector distance was increased from 10 to 50 mm with increments of 10 mm (Table 4.2). In the last set, we increased the flow rate from 2 to $5\,\mathrm{mL\,h^{-1}}$ with $1\,\mathrm{mL\,h^{-1}}$ increments while keeping the other two process parameters constant (Table 4.3). PLA material, with properties listed in Table 4.4, was used for all three simulations. In addition, the remaining processing parameters were set to values listed in Table 4.5. Each simulation was run for a total of 0.025 s.

Table 4.1: Effect of applied voltage on fiber morphology. (distance = 50 mm, flowrate = $2\,\mathrm{mL\,h^{-1}}$)

Applied voltage (kV)	Fiber diameter (µm)	Collection diameter (mm)	Maximum speed ($\mathrm{m\,s^{-1}}$)	Percentage elongation (%)
5.00	8.52	10.36	4.80	452
10.00	8.34	9.05	6.52	476
15.00	8.58	8.16	7.86	443
20.00	7.61	6.61	9.02	590

4.3.1 Observations Common to the Three Simulation Cases

A quick inspection of the results indicated the presence of straight jets from the nozzle tip to the collector plate. However, scaling up the x and y axes displacement

Table 4.2: Effect of collector distance on fiber morphology. (voltage = 10 kV, flowrate = 2 mL h^{-1})

Collector distance (mm)	Fiber diameter (µm)	Collection diameter (mm)	Maximum speed (m s^{-1})	Percentage elongation (%)
10.00	10.12	2.34	6.45	291
20.00	9.86	4.00	6.46	312
30.00	8.76	5.18	6.47	421
40.00	8.53	6.57	6.49	450
50.00	7.80	8.15	6.51	557

Table 4.3: Effect of volumetric flowrate on fiber morphology. (voltage = 10 kV, distance = 50 mm)

Flow rate (mL h^{-1})	Fiber diameter (µm)	Collection diameter (mm)	Maximum speed (m s^{-1})	Percentage elongation (%)
2.00	8.12	7.35	6.51	507
3.00	9.14	8.48	6.53	379
4.00	8.88	9.71	6.54	408
5.00	9.13	8.86	6.56	379

Table 4.4: Melt material parameters

Parameter	**Value**
Density (melt)	1073 kg m^{-3}
E_0	1.52 MPa
E_1	4.20 MPa
η_1	1320 kg/(ms)
κ_t	1×10^{-12} N m rad^{-1}
c_t	1×10^{-12} N m rad^{-1}
Surface charge density	29.3×10^{-6} C m^{-2}

Table 4.5: Process parameter settings

Parameter	**Value**
Melt temperature	185 °C
Ambient/enclosure temperature	60 °C
Nozzle diameter	0.5 mm
Spinning configuration	Vertical

values showed that the jets were, in fact, characterized by subtle whipping motions that grow in amplitude towards the collector plate. As evidenced by Fig. 4.1.b, the relative straightness in the jet flow is one of the characteristic traits of MES.

The jets also went through a thinning phase, indicated by the growing spacing gap between successive bead markers as the jet traveled toward the collector plate. This was followed near the collector plate by a stage wherein the distances between successive beads seemed unchanged. This is consistent with the rapid fiber thinning and eventual solidification observed in both SES and MES [385]. This can be explained in terms of rheology by the changes in the material's elastic properties in response to a temperature drop from the melt temperature inside the nozzle to the ambient temperature at the collector plate [305]. The values of both viscosity and modulus of elasticity are reported in the literature to increase with logarithmic rates in response to temperature drop [386–389]. The rate of viscosity increase, however, is much faster than that of the rate of change in modulus of elasticity, resulting in increased relaxation time as the jet approaches the collector plate. This forces the elongated material to retain its deformation for an extended period of time, leading to the jet's solidification before it gets a chance to recover from the deformation.

The jets started with negligible initial velocities at the nozzle tip in all three cases. They accelerated towards the collector plate, eventually reaching final speeds of up to 9 m/s right before adhering to the collector plate.

4.3.2 Effect of Applied Voltage, Collector Distance, and Flowrate on Fiber Morphology and Topology

While the final fiber diameter and the diameter of the collected fiber mat decreased in response to the increased applied voltage (Table 4.1), the magnitudes of the jet speed and the fiber elongation percentage increased. This observation can be explained by the fact that increasing the applied voltage increases the electric field forces on the beads, making the pulling force toward the collector plate more prominent than the other forces. Hence more fiber stretching and increased acceleration are observed in the direction of this force.

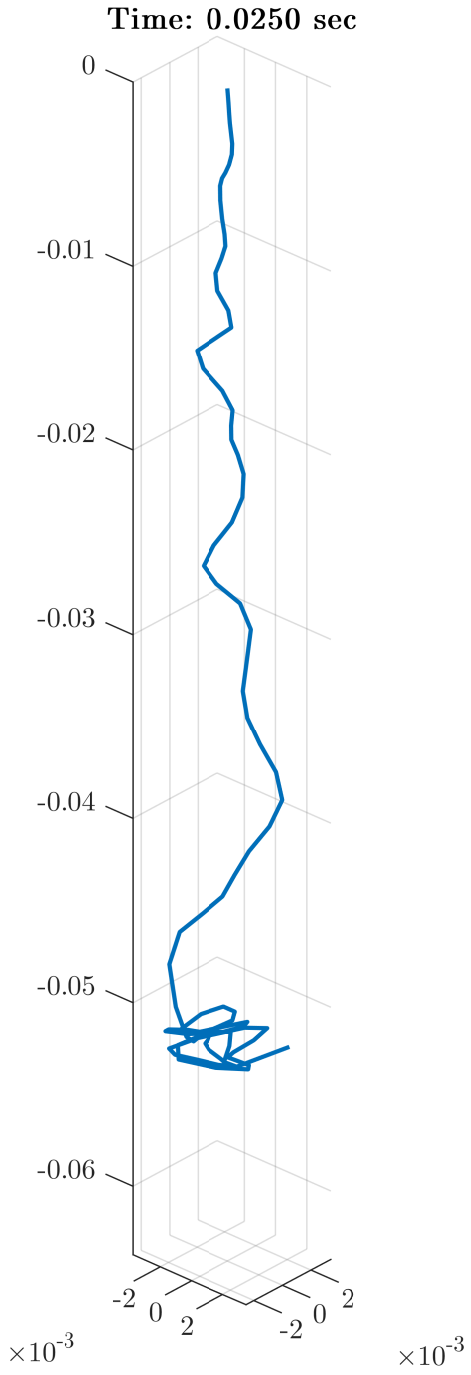


Figure 4.9: Vertical MES simulation result

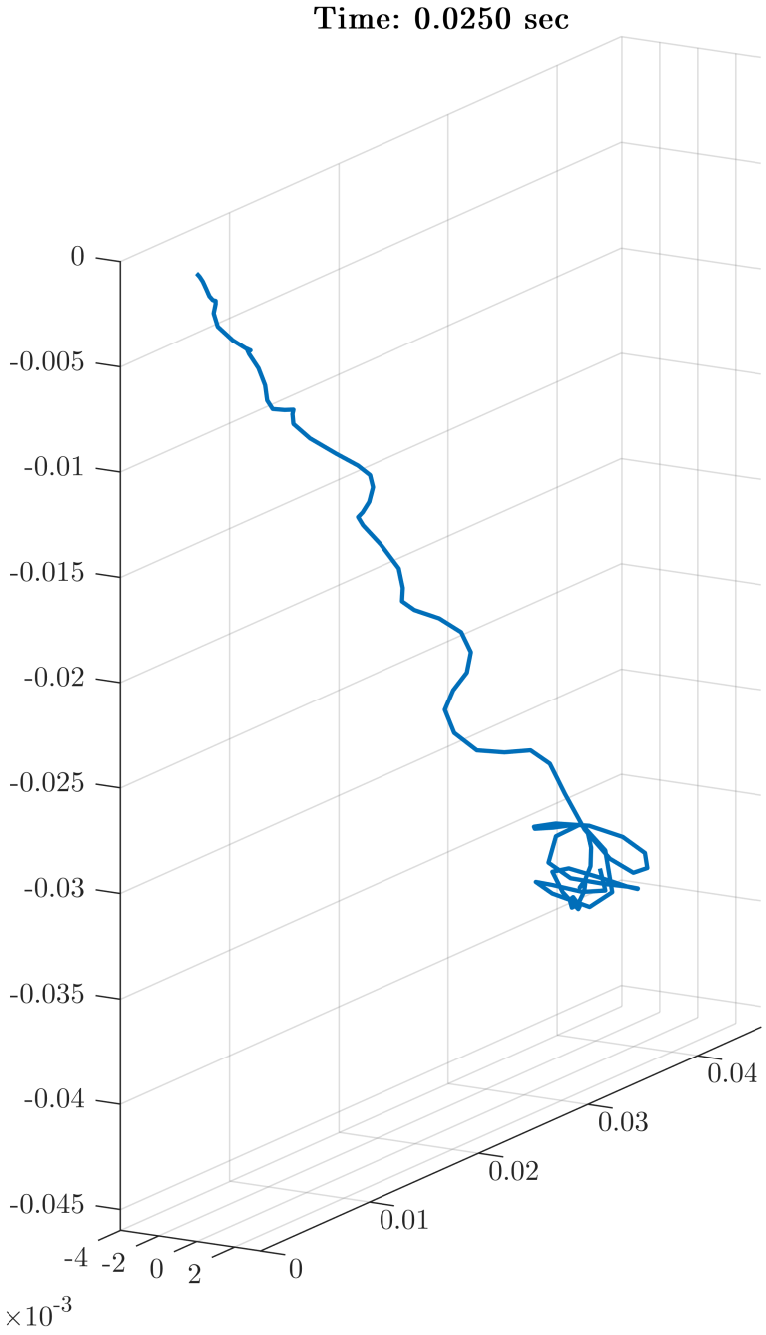


Figure 4.10: MES simulation at a 45° spinning angle

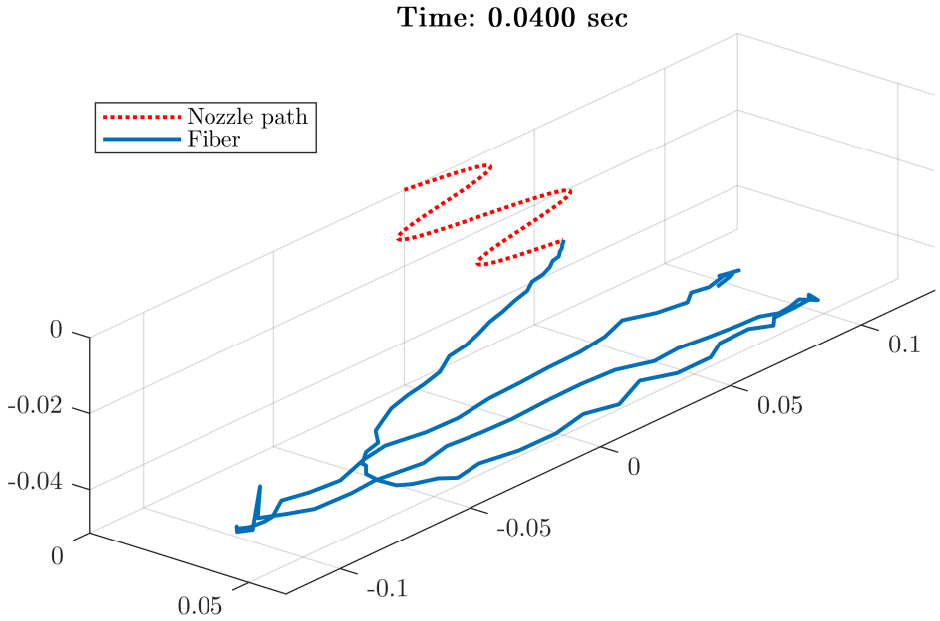


Figure 4.11: Vertical MES with nozzle moving in a prescribed sinusoidal path

Data from Table 4.2 shows that while the fiber diameter decreased in response to increasing the collector distance, the remaining three output parameters increased in magnitude. Because increasing the collector distance has the same effect as weakening the electric field strength, we anticipated seeing an increase in the fiber diameters (*i.e.*, less thinning). However, the results indicated a more prominent fiber diameter reduction than the case of voltage increase, indicating that the less-restricted whipping motion itself might have played a role in thinning the fiber. Also, in line with our expectation, an increase in the collector distance was accompanied by an increase in the collected mat diameter, confirming the already known fact that a more ordered fiber topology is achieved near the stable region.

Measurements in Table 4.3 showed a direct relationship between the flow rate and the fiber diameter. An increase in the fiber diameter (cross-sectional area) means an increase in the viscoelastic force, which, in turn, creates additional resistance to fiber deformation. This resistance leads to the reduction in elongation observed in the same results table. The jet speeds seem to increase in a minimal magnitude in response to increasing the flow rate. However, we believe this has more to do with an uptick in the initial velocity at the nozzle due to the increased flow rate and less

to do with contributions by the motion's dynamics.

4.3.3 Further Considerations

Extremely fast motions characterize the fiber dynamics in the model. As such, a comparatively short time step (in the range of 2.85×10^{-5} s) had to be selected for the simulations to achieve reasonable accuracy. Failing to select this time step appropriately could result in errors, which, sometimes, is evidenced by sharp bends in the simulation results. This challenge is a concern because, according to the algorithm in Fig. 4.8, the last solution of each iteration is used as an initial condition in the next, and the error introduced at each stage could easily stack up and lead to an incorrect result. However, the extremely short time step also meant we could quickly generate a large amount of data. For example, running a five-second simulation with the above time step would generate $5\,\text{s}/2.85 \times 10^{-5}\,\text{s} = 175439$ rows of data. On most 64-bit computers, this data size is outside of available memory. For this reason, and because not much new information is generated after the jet quickly reaches a steady state, we have decided to limit the simulation time to 0.025 s. As shown in Fig. 4.9 – 4.11, this duration was long enough to generate sufficient data for sensitivity analysis.

Another way this challenge could be addressed is by simulating multiple sessions using appropriately short-time steps that guarantee smooth fiber motion profiles. Because of the shorter time duration of each session, however, one might not be able to draw a meaningful conclusion from a single session's result. However, patching together the results could provide a better picture. For this to work, a given session's result is saved as a separate file, with its last iteration step results (states) used as initial conditions in the next session. This process is repeated until the total simulation length is reached.

For the same reason, we cannot create one big simulation file in a single session (memory issue) and patch together the individual files to create the complete time history. However, a less memory-intensive interpretation of the results could be extracted from each solution file and patched together to provide some crucial insights into the overall process. For example, instead of focusing on the animation (which is a memory-intensive process), only the final profiles of the collected fiber can be

extracted, which are static 3D curves (less memory).

Parameter estimation was another area that was identified as needing more improvement. A generalized Maxwell viscoelastic model, with at least 10–15 Maxwell elements, is needed to match test data collected through DMA and rheological tests [28]. In contrast, the single-layer standard Maxwell viscoelastic solid (SLS) model used in the present work, although still an improvement compared to existing models, is still inefficient in representing the material properties. However, from a computational standpoint, it makes sense to keep the order of differentiation of the stress and strain terms as low as possible to avoid errors arising from improper initial condition estimation. For example, the single-layer SLS model's stress variable σ can be estimated with reasonable accuracy from the initial fiber diameter, generalized coordinates, and generalized speeds. In contrast, if a second layer is present, initial values for both σ and its time derivative, $\dot{\sigma}$, would need to be estimated, with the latter becoming more challenging to handle. Similarly, a third layer would require initial value estimation for σ, $\dot{\sigma}$, and $\ddot{\sigma}$. The larger the number of layers in the SLS (higher order derivatives), the more challenging to estimate the initial conditions for the stress terms. Because of these concerns, we have not attempted to simulate beyond a single-layer SLS.

4.4 Conclusion

This study presents an analytical model representing the dynamics of a melt-electrospun fiber as a serially connected multi-body system. The generalized coordinates and velocities were defined to describe the spatial configuration and kinematics of the beads. Forces and torques in the system were identified and successfully implemented into the system via their explicit mathematical expressions. Partial velocities and partial angular velocities expressions were derived for the beads using Kane's method and later used in formulating the unconstrained equations of motion of the system. The constraint forces were systemically formulated and later imposed on the unconstrained equation of motion using the Udwadia-Kalaba method, leading to the fully constrained equations of motion. A brief discussion of the analytic schemes for the computer implementation of the proposed model was also presented. In particular,

an algorithm that enables continuous simulation of the system was proposed. The resulting equations of motion and the algorithm were implemented via a commercial numerical integration package (*i.e.*, ode45 from MATLAB®).

Three representative scenarios were simulated to investigate the relationships between the selected MES process and output parameters. The effects of changes in applied voltage, collector distance, and melt flow rate on the morphological and topographical characteristics of melt-electrospun fibers were individually investigated. The results were presented with thorough discussions. The main results can be summarized as; 1) increasing voltage reduces the fiber diameter and the collected mat diameter, 2) increasing collector distance results in a smaller fiber diameter while at the same time giving rise to an enlarged collector mat diameter, and 3) increasing melt flowrate has the effect of increasing both the fiber diameter and the collector mat diameter.

We have successfully demonstrated the unique features of the present model, including its ability to continuously simulate the dynamics of the fiber, with the motions of the nozzle and the collector plate taken into consideration. The effects of the fiber sticking onto the collector plate on the motion profiles of the fibers have also been examined via the proposed numerical scheme.

Lastly, the proposed model and numerical schemes presented may be employed to investigate the topological characteristics of melt electrospinning in direct-writing and 3D printing applications. The data obtained from the model can be further processed to estimate spatial distributions of collection density and printing resolutions, which are critical in filtration and biomedical applications. The ability to predict fiber characteristics could be instrumental in bone scaffold fabrications, wherein inter-fiber spacing partly influences the type of bone tissue growth.

Chapter 5

Experimental Characterization of Melt-Electrospinning and Theoretical Validation of Predictive Model

Chapter Abstract

The current study extends a previous work published by the authors in which an analytical predictive model is proposed to simulate the melt-electrospinning (MES) process. The analytical model is specifically designed to predict the various behaviors of the melt-electrospun fiber under different material and processing conditions. A brief discussion of this model is presented here to establish context and help the reader capture the modeling philosophy employed. The current study complements the previous work by focusing on the experimental aspects of the research. Correlations between the independent process parameters and the topological attributes of the melt-electrospun fibers are investigated and compared with findings from the theoretical model. The effects of changes in the process parameters on average fiber diameters and the collection diameter are experimentally analyzed using the design of experiments (DOE) techniques. Towards this end, polylactic acid (PLA) is melt-electrospun at different treatment levels of the processing parameters in a controlled environment. Two regression-based models—one for predicting the collec-

tion diameter and the other for the fiber diameter—are derived from the DOE data for benchmarking and quantitative evaluation of the predictive performance of the theoretical model. The theoretical model is run based on the same treatment levels as the experiment. The elastic parameter values used in the theoretical simulation are extracted from rheological tests. Comparison between the simulated and the observed fiber characteristics revealed that the collector diameter predictions by the theoretical model exhibited approximately a 16.7% difference compared to 24.2% for the average fiber diameter. Finally, a discussion is presented on the challenges and potential factors contributing to the observed differences. Overall, given the identified challenges and gaps in material characterization, the results of the theoretical predictive model are encouraging.

5.1 Introduction

Recent advancements in material science and fabrication technologies have presented exciting opportunities in the field of tissue engineering [19, 20, 390]. MES, which is the process of drawing fibers from an electrically charged molten polymer, has become the focus of a growing number of studies [2]. This is partly because the process lends itself to various applications, including textile manufacturing, filtration systems, and biomedical applications. One particular application area of interest has been integrating MES with three-dimensional (3D) printing technologies for guided tissue regeneration purposes [202, 227, 391, 392]. The close resemblance and functional compatibility between the intra-fiber micro-architecture of the melt-electrospun fiber accumulations and the extracellular matrix (ECM) of the natural bone tissue make the process an ideal candidate for the fabrication of bone scaffolds. Some clinical trials have already demonstrated direct correlations between the type of vascularization observed during the bone regeneration process and, among several other factors, the geometrical architecture of the scaffolds tested [22, 23]. Stringent clinical requirements in these applications underscore the need for proper techniques to characterize and control the process. For example, the average fiber diameter, the collection diameter, and the density of melt-electrospun fiber are all influenced by the material

properties, processing parameters, and other environmental factors such as ambient temperature and humidity. Therefore, a comprehensive understanding of the effects of the processing parameters on the dynamic behavior of the electrospinning process is essential. It is a critical step toward building bone scaffolds with targeted vascularization properties.

Different groups in the past have proposed theoretical and experimental models to describe various aspects of solution-electrospinning (SES) and MES processes [297, 298, 353–357]. For example, the critical voltage level required to trigger the jet stream in a solution-based electrospinning process was first estimated by Taylor [346]. Others later expanded his work to include the characterization of the polymer jet flow behaviors in the stable regions [314, 347, 348]. Reneker *et al.* [315] was the first to propose a model for predicting the motion profiles of the electrospun fibers in the bending instability regions. They treated the fiber mathematically as a serially connected multi-body system joined by a simple Maxwell viscoelastic link (spring-dashpot connection). On the experimental front, Dalton *et al.* [279, 351] have experimentally characterized the relationships among the processing parameters of MES and the resulting fiber topology. They also investigated the application of the process in direct writing, wherein the melt-electrospun fiber is deposited in a predefined and highly ordered manner. Their work has provided valuable insights into the effects of translational nozzle speeds on the fibers' accumulation patterns and other topological characteristics.

The current work is an extension of these modeling efforts. It builds upon a prior work published by the same authors in which a theoretical predictive model was proposed for MES [393]. The relationships between the input and output parameters are experimentally investigated under the current work to draw a comparison to this theoretical model. In addition, the experimental data is also used to construct predictive regression models to provide a means for equivalent comparisons and objectively assess the predictive performance of the theoretical model.

5.2 Brief Background on the Theoretical Model

The theoretical model is developed with the primary objective of simulating the dynamic motion of the fibers during the MES process. The simulation data generated is further analyzed to extract critical characteristics of the collected fibers. The strategy employed in the development of the analytical model follows that of Reneker's approach [315] in that the fiber is mathematically treated as a serially connected multi-body system composed of discrete fiber segments (Fig. 5.1). Each segment is treated as a rigid body (bead) with inertial and electrical charge properties. A set of equations of motion is first developed for each bead by considering the effects of various forces acting on it, including its interaction with neighboring beads. The forces considered include gravitational, Coulomb, viscoelastic, surface tension, and electric field forces.

While the similarities are limited to these, there are some critical differences and several advanced improvements made in our model [393]. First, Maxwell's generalized *standard linear solid (SLS)* viscoelastic element (Fig. 5.2) is employed instead of the simple spring-dashpot link found in many of the previous models. The added extra layer in this approach provides additional degrees of freedom that improve the accuracy of material representation when extracting the elastic parameters of the material from a rheological test. Second, an additional spring-dashpot element is also applied in the rotational sense to account for viscoelastic bending deformations (Fig. 5.1). While the SLS serves as a linear connective element between the beads, the rotational viscoelastic element provides elastic and damping bending resistances.

The other significant improvement is that, unlike previous models in the literature, the current model is designed to account for the eventual touching down and sticking of the fiber onto the collector plate (*anchoring effect*). This improvement is achieved by defining the plane of the collector plate as a holonomic constraint and using it as an event trigger in the numerical integration. In addition, to account for the potential use of the predictive model in simulating MES in 3D printing applications, the dynamic contributions of motions of the nozzle and the collector plate are also included in the analytical problem formulation in the form of additional holonomic

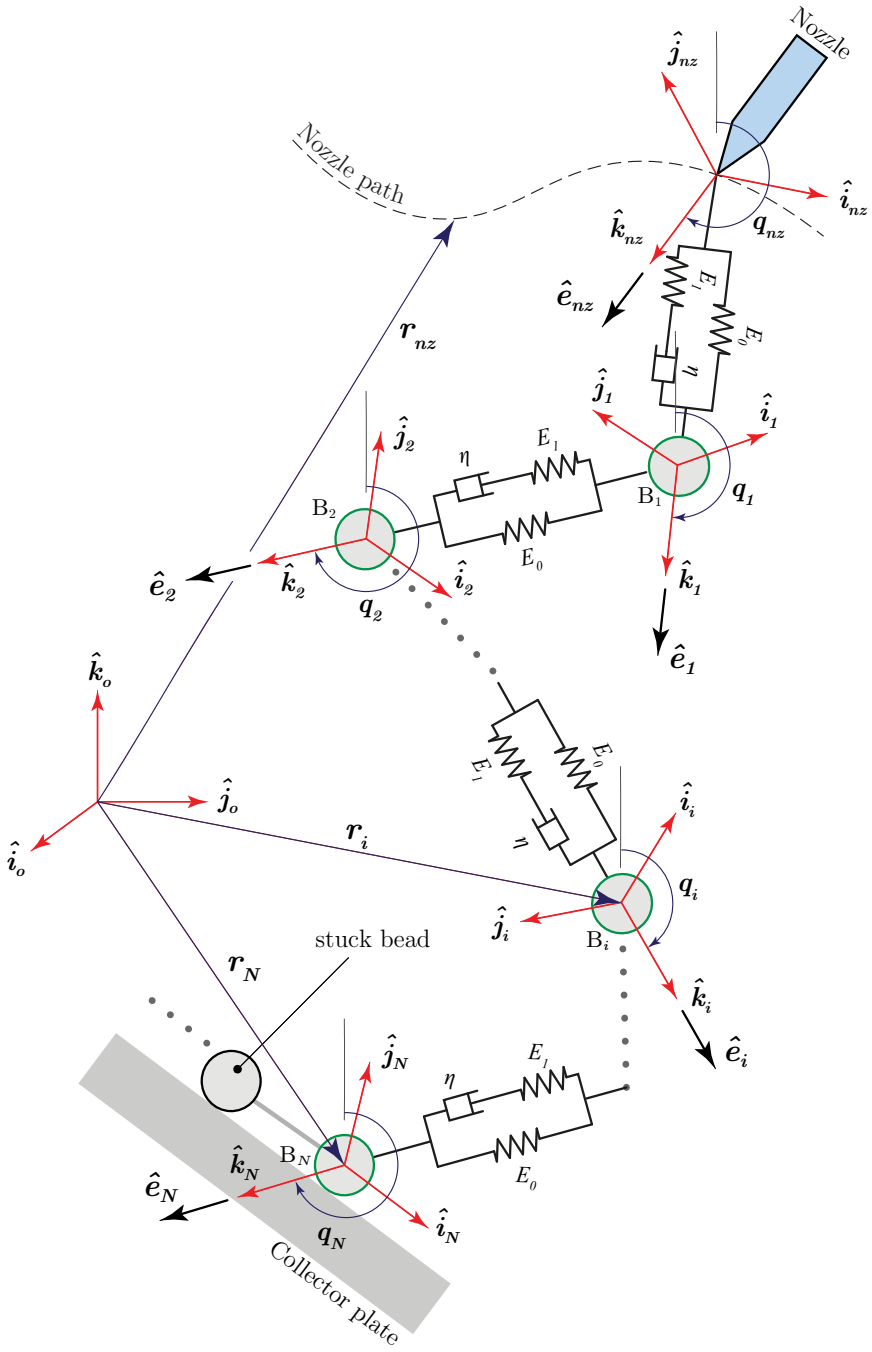


Figure 5.1: Conceptualization of melt-electrospun fiber as a multi-body system

constraints.

After selecting Cartesian coordinates and unit quaternions describing the position and orientation, respectively, of each bead as generalized coordinates, the unconstrained dynamical equations of motions of the system are derived using Kane's method [342, 358, 394–396]. The generalized coordinates, the tensile stress $\sigma(t)$ in Fig. 5.2, and their respective time derivatives constitute the system variables in the resulting equations of motion. Holonomic constraint equations arising from the quaternion identity, the fiber-plate sticking phenomenon, and the nozzle's relative motion are implemented using the Udwadia–Kalaba method [338, 359–362]. These constraints are later appended to the unconstrained equations of motion to furnish the constrained equation of motion of the fiber as a multi-body system. The resulting system of differential-algebraic equations (DAE) is numerically solved in MATLAB®using the ode45 solver to predict the time history of each bead in the system.

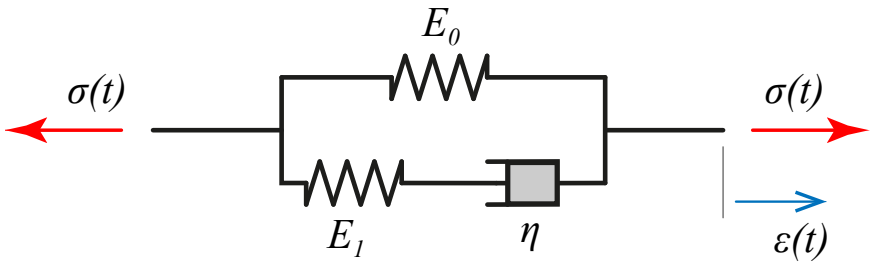


Figure 5.2: Generalized viscoelastic element

For the single-layer SLS model shown in Fig. 5.2 (*i.e.*, $N = 1$), the relationship between its stress and strain terms and the elastic parameters of the material is governed by the constitutive equation:

$$\sigma + \frac{\eta_1}{E_1}\frac{\partial \sigma}{\partial t} = E_0\epsilon + \frac{\eta_1}{E_1}(E_0 + E_1)\frac{\partial \epsilon}{\partial t}. \tag{5.1}$$

Viscoelastic forces on the fiber are calculated by multiplying the stress σ variable by the cross-sectional area of the fiber. This constitutive equation, however, becomes more complicated as the number of layers in the generalized model increases. For example, for a two-layer generalized viscoelastic model, *i.e.*, $N = 2$, the equation

takes the form:

$$\sigma + \left(\frac{\eta_1}{E_1} + \frac{\eta_2}{E_2}\right)\frac{\partial\sigma}{\partial t} + \frac{\eta_1}{E_1}\frac{\eta_2}{E_2}\frac{\partial^2\sigma}{\partial t^2} = E_0\epsilon + \left(\frac{\eta_1}{E_1}(E_0+E_1) + \frac{\eta_2}{E_2}(E_0+E_2)\right)\frac{\partial\epsilon}{\partial t} + (E_0+E_1+E_2)\frac{\partial^2\epsilon}{\partial t^2}, \quad (5.2)$$

which has second-order partial differential terms of the stress σ and strain ϵ variables. The equation becomes even more complicated and too long to print here for models containing three or more layers. Generally, the constitutive equation of an SLS element with N layers will have $\frac{\partial^N\sigma}{\partial t^N}$ and $\frac{\partial^N\epsilon}{\partial t^N}$ terms. This poses a serious challenge arising from difficulties in estimating initial conditions for the higher-order terms during numerical integration. Due to the chaotic nature of MES, the solution to the dynamical equations are highly sensitive to the initial conditions. Hence, roughly estimated initial values of the higher-order terms could lead to an erroneous outcome. Avoiding this challenge makes lower numbers of layers a justifiable and necessary option. However, a lower number of layers also means sub-optimal curve-fitting and poor material representations, which could also translate into a loss of accuracy. Nevertheless, the latter approach has the added benefit of resulting in less complex equations of motion, which run much more efficiently during computer simulations.

5.3 Methodology and Materials

5.3.1 Equipment

A commercially available 3D printer (Ender 5 Pro from Creality) was acquired and modified into a MES apparatus (Fig. 5.3). The modifications included; installing a high-voltage power supply (*Gamma High Voltage Research, ES30*), installing high-voltage electrical insulation, adding a custom flow rate controller and software, building an enclosed printing chamber, and installing a chamber heater and a temperature controller unit.

5.3.2 Materials

A 1.75 mm diameter Polylactic acid (Ecogenius PLA 3D, Sigma-Aldrich) filament was used in the experiment. The flow rate ($\dot{Q}$), applied voltage (V), collector distance

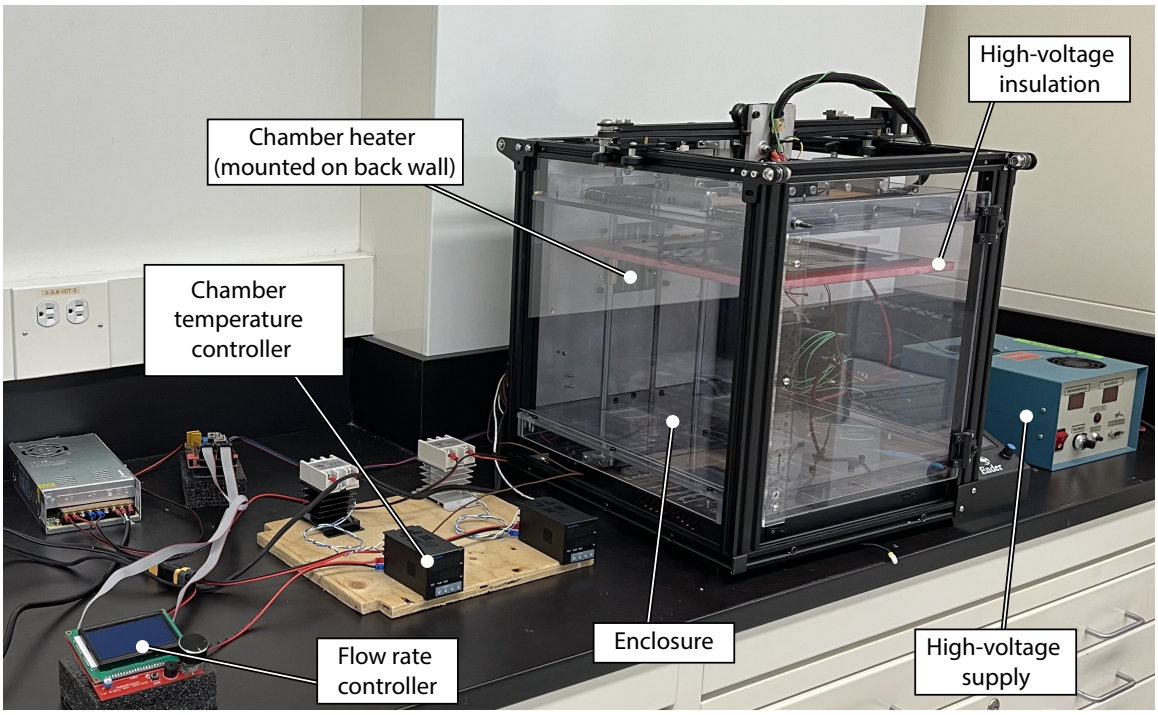


Figure 5.3: Melt electrospinning apparatus

(h), and melt temperature (T_m) were selected as independent variables (factors). Table 5.1 is a list of three discrete levels for each of these factors, carefully selected in the neighborhood of an optimal morphological processing point that had been determined from a previous experiment ($\dot{Q}^* = 2\,\mathrm{mL\,h^{-1}}$, $V^* = 17.5\,\mathrm{kV}$, $h^* = 40\,\mathrm{mm}$, and $T_m^* = 240\,^\circ\mathrm{C}$).

5.3.3 Rheological Characterization of PLA

Frequency sweeps of the storage modulus G', loss modulus G'', and complex viscosity η^* of the PLA material were collected from a rheological test (AR2000 Rheometer, Texas Instruments, USA). The rheological data was later curve-fitted using Eq. (5.3) and Eq. (5.4) to estimate the elastic parameters of the material (E_0, E_1, and η_1) using similar procedures described in the literature [28]. This procedure was repeated at

the three melt temperature points to find the corresponding elastic parameter values.

$$G'(\omega) = G_0 + \sum_{1}^{N} g_k \frac{(\omega\tau_k)^2}{1 + (\omega\tau_k)^2}, \tag{5.3}$$

$$G''(\omega) = \sum_{1}^{N} g_k \frac{(\omega\tau_k)}{1 + (\omega\tau_k)^2}, \tag{5.4}$$

where τ_k is the relaxation time calculated from $\tau_k = \eta_k / E_k$.

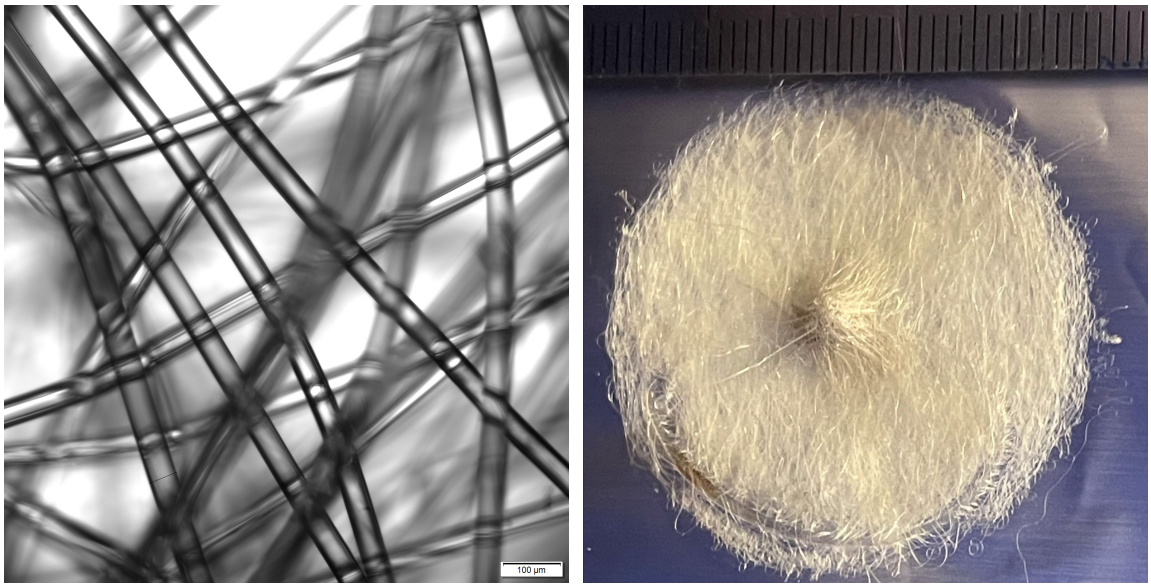
Table 5.1: Factors and levels used in the simulations and experiment

Factor	**Level 1**	**Level 2**	**Level 3**
Flow rate ($\mathrm{mL\,h^{-1}}$)	1.0	2.0	3.0
Applied voltage (kV)	15.0	17.5	20.0
Collector distance (mm)	30.0	40.0	50.0
Melt temperature (°C)	230.0	240.0	250.0

The theoretical model and the experiments were run using the same material and processing parameter values. The full-factorial experiment was conducted inside an enclosed chamber heated to 60 °C based on the factors and levels listed in Table 5.1. Each physical sample was collected over two minutes after stabilizing the process around its set point. The observed fiber diameter (d_f) for each sample was calculated as the average of 35 measurements taken in ImageJ software (Fig. 5.4(a)). On the other hand, the observed collection diameter (d_c) was reported as a one-time measurement of the diameter of an inscribing circle drawn around the collected sample.

For a given simulation, the theoretical collection diameter ($\tilde{d}_c$) was computed as the maximum distance measured among the stuck beads on the plane of the collector plate. Whereas, the theoretical fiber diameter ($\tilde{d}_f$) was calculated as the average of all fiber diameters between successive bead pairs.

Two regression models—one for the prediction of the collection diameter ($\hat{d}_c$) and another for the average fiber diameter ($\hat{d}_f$)—were derived from the experimental data for benchmarking purposes. The predictive performance of the theoretical model was evaluated by comparing its outputs with the corresponding outputs of the regression models under the same material property and processing conditions (factor levels).



(a) (b)

Figure 5.4: Melt-electrospun fibers (PLA): (a) Magnified with bifocal microscope; (b) Without magnification

The residuals of the theoretical and regression model predictions were statically analyzed and reported as performance metrics.

5.4 Results and Discussions

5.4.1 Material Characterization

First, frequency sweeps of the complex viscosity, storage modulus, and loss modulus of PLA were collected from rheological tests. Figures 5.5(a)–(c) show the data collected from these tests at three temperatures levels (*i.e.*, 230 °C, 240 °C, and 250 °C). The plot trends indicate reductions in the viscosity and the moduli by logarithmic order of magnitudes in response to increased melting temperatures. However, in the actual MES process, the electrospun fibers are expected to drop their temperatures once they exit the spinneret and experience the opposite trend; the magnitudes of viscosity and the moduli increase as the fibers cool down in the printing chamber.

Next, the raw rheology data in the graphs were curve-fitted using Eq. (5.3) and (5.4) to determine the elastic parameters of the material (E_0, E_i, and η_i). As demonstrated in Figs. 5.5(d) and 5.5(e), the number of Maxwell layers used in the model affect the

accuracy of the curve fitting. More layers in the model mean more elastic parameters in the resulting viscoelastic constitutive equation and better viscoelastic behavior representation. The estimated elastic parameter values corresponding to the different numbers of layers (*i.e.*, $N = 1, 2, 15$) are summarized in Table 5.2. The data used for the parameter estimation was collected at 200 °C.

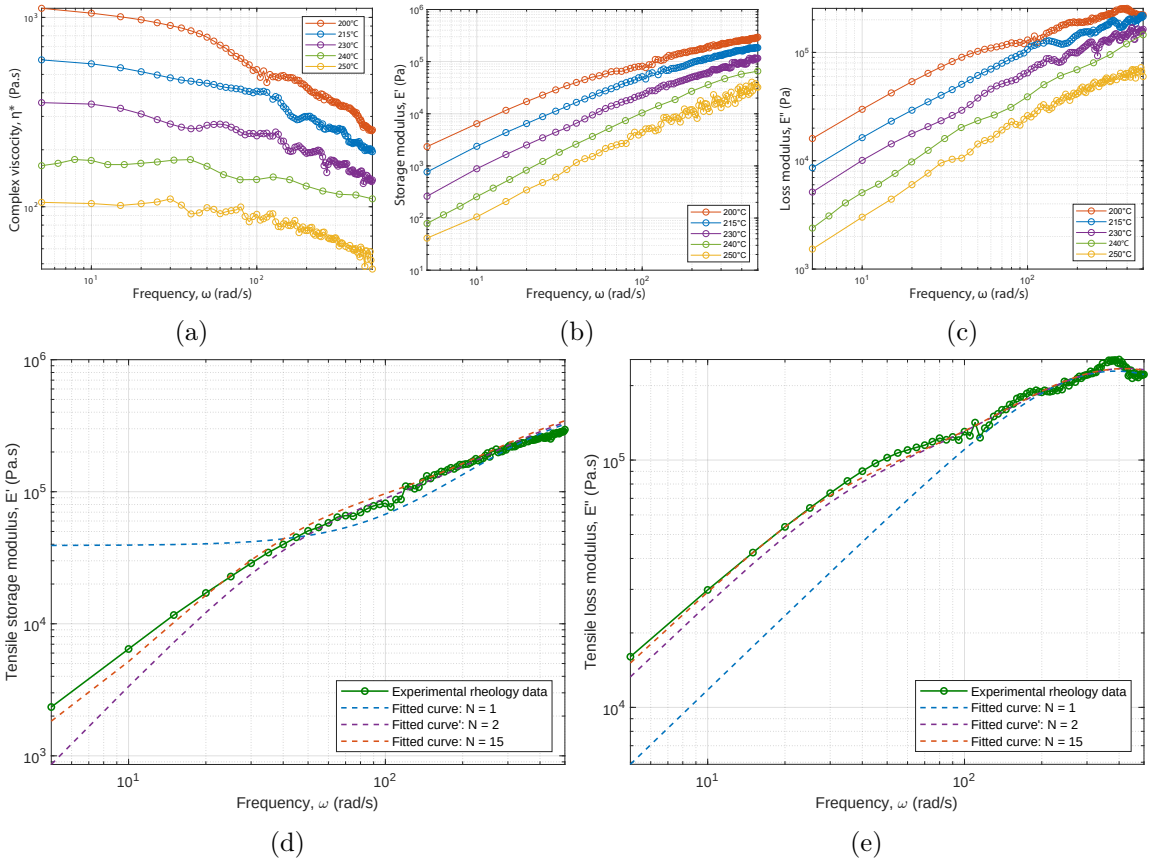


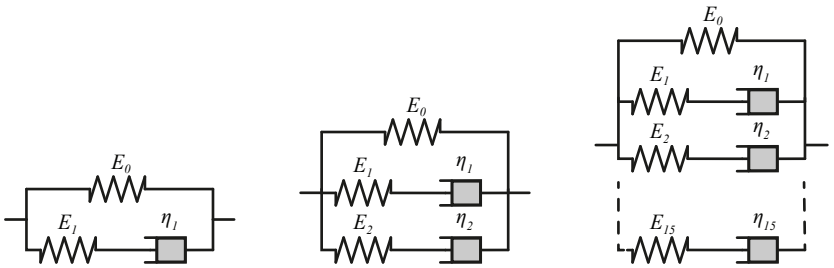
Figure 5.5: Rheological testing and elastic parameters estimation of PLA: (a) Complex viscosity vs. frequency of PLA; (b) Storage modulus vs. frequency of PLA; (c) Loss modulus vs. frequency of PLA; (d) Storage modulus of PLA at 200 °C fitted with Maxwell's viscoelastic models of different layer numbers; (e) Storage modulus of PLA at 200°C fitted with Maxwell's viscoelastic models of different layer numbers

5.4.2 Regression-based Predictive Models

Appropriate regression modeling schemes are needed to fit the experimentally observed d_c and d_f response data. Different regression models, such as two-factor interaction, quadratic, and cubic, were investigated for this purpose. Based on these comparisons, the cubic polynomial modeling scheme, slightly modified by removing terms with a *p-value* greater than 0.05, was chosen for both the collection diameter

Table 5.2: Estimated elastic parameter values at $T_m = 200\,^{\circ}\mathrm{C}$

	N = 1		N = 2		N = 15	
Layer	E (Pa)	η (Pa s)	E (Pa)	η (Pa s)	E (Pa)	η (Pa s)
0	39121.53		0		47.15	
1	457613.1	1177.05	449379.13	1009.22	207.37	143.96
2			85291.72	1662.86	392.17	234.23
3					0.49	283.32
4					0	518.28
5					14721.28	330.59
6					42.08	115.02
7					25020.2	561.86
8					279861.44	645.11
9					20774.82	466.5
10					27433.43	616.08
11					0.03	95.34
12					0.05	996.71
13					42.07	114.69
14					62533.22	144.21
15					107753.87	247.8



and the average fiber diameter response data. The analysis of variance (ANOVA) in Table 5.3 and Table 5.4 show the relative significance of factors affecting the collection diameter and the average fiber diameter responses, respectively. Factors with smaller *p-value* (< 0.0001) have strong direct correlations with the corresponding responses. For example, the $\dot{Q}$, h, T_m, and h^2 factors in Table 5.3 have strong effects on the collection diameter. These strong correlations were also brought to evidence by the one-factor-at-a-time plots of interactions in Fig. 5.7(c)–(f) and Fig. 5.9(c)–(f). Equation (5.5) and Eq. (5.6) are the corresponding regression formulas for the collection and average fiber diameter observed responses.

$$d_c = 679.837 + 2.091\dot{Q} - 64.210V - 4.483h - 3.121T_m + 0.135Vh + 0.300VT_m + 0.030hT_m + 1.444V^2 - 0.001h^2 - 0.001VhT_m + 0.006V^2h - 0.006V^2T_m, \quad (5.5)$$

Table 5.3: ANOVA table for collection diameter

Source	Sum of Squares	DoF	Mean Square	F-value	p-value
Model	523.18	12	43.6	67.91	<0.0001
$\dot{Q}$ – Flowrate	225.89	1	225.89	351.85	<0.0001
V – Applied voltage	0.0739	1	0.0739	0.1151	0.7355
h – Collector distance	54.71	1	54.71	85.21	<0.0001
T_m – Melt temperature	30.71	1	30.71	47.83	<0.0001
Vh	0.045	1	0.045	0.0702	0.7919
VT_m	0.2759	1	0.2759	0.4297	0.5144
hT_m	3.63	1	3.63	5.65	0.0203
V^2	0.9808	1	0.9808	1.53	0.2208
h^2	5.68	1	5.68	8.84	0.0041
VhT_m	3.44	1	3.44	5.36	0.0237
V^2h	1.95	1	1.95	3.04	0.0861
V^2T_m	2.24	1	2.24	3.48	0.0665
Residual	42.37	66	0.642		
Correlation total	565.55	78			

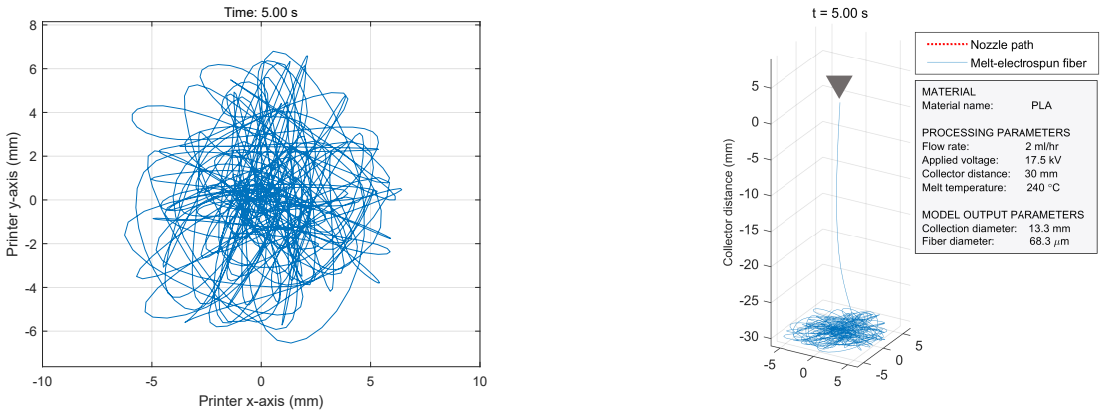
Table 5.4: ANOVA table for average fiber diameter

Source	Sum of Squares	DoF	Mean Square	F-value	p-value
Model	36816.5	10	3681.65	45.67	<0.0001
$\dot{Q}$ – Flowrate	16104.47	1	16104.47	199.77	<0.0001
V – Applied voltage	3369.93	1	3369.93	41.8	<0.0001
h – Collection distance	1714.5	1	1714.5	21.27	<0.0001
T_m – Melt temperature	12342.93	1	12342.93	153.11	<0.0001
$\dot{Q}V$	44.93	1	44.93	0.5574	0.4579
$\dot{Q}T_m$	1634.27	1	1634.27	20.27	<0.0001
VT_m	268.77	1	268.77	3.33	0.0723
$\dot{Q}^2$	2.01	1	2.01	0.0249	0.8751
h^2	520.37	1	520.37	6.46	0.0134
$\dot{Q}^2V$	1098.55	1	1098.55	13.63	0.0004
Residual	5481.71	68	80.61		
Correlation total	42298.21	78			

$$d_f = 462.525 + 460.050\dot{Q} - 15.735V - 4.945h - 2.104T_m - 15.956\dot{Q}V - 0.685\dot{Q}T_m + 0.111VT_m - 67.446\dot{Q}^2 + 0.054h^2 + 3.873\dot{Q}^2V. \quad (5.6)$$

5.4.3 Collection Diameter Prediction Performance

Based on the set of factors and levels listed in Table 5.1, 81 treatments and simulations were run using the regression and theoretical models. The complete experiment data is provided in Appendix B. The experiment was followed by comparisons of the collection and fiber diameters predicted by the models with their observed counterparts. Figures 5.6(a)–(b) show the top and perspective views of the simulation results at the end of 0.5 s.



(a) (b)

Figure 5.6: Sample simulation result: (a) Top view of simulation output, (b) Perspective view with simulation summary

Figures 5.7(a) and 5.7(b) show the distributions of the collection diameters predicted by the regression-based and the theoretical models, respectively, against their observed counterparts. The 45° reference lines in the plots represent the ideal cases when the predicted and observed values match. The closer the data points lie to these reference lines, the better the accuracy of the predictive models. The data points in Fig. 5.7(a) are proportionally and closely dispersed on both sides of the 45° reference line, indicating that the regression-based collection diameter prediction model has reasonable accuracy and a normal distribution of the residuals. The residuals were calculated as the difference between the predicted observed values, leading to 81 residual points for the regression-based and theoretical models each (*i.e.*, regression: $\hat{\Delta}_{ci} = \hat{d}_{ci} - d_{ci}$, theoretical: $\tilde{\Delta}_{ci} = \tilde{d}_{ci} - d_{ci}, i = 1, ..., 81$)

The predictive performances of the regression model and the theoretical model were evaluated by statistically comparing their residuals. The raw residuals were first converted to percentage residuals by simply dividing the raw residuals by the corresponding observed values, *i.e.*,

$$\hat{\Delta}_{ci} = 100 * |d_{ci} - \hat{d}_{ci}|/d_{ci}, \tag{5.7}$$

$$\tilde{\Delta}_{ci} = 100 * |d_{ci} - \tilde{d}_{ci}|/d_{ci}, \tag{5.8}$$

$$\hat{\Delta}_{fi} = 100 * |d_{fi} - \hat{d}_{fi}|/d_{fi}, \tag{5.9}$$

$$\tilde{\Delta}_{fi} = 100 * |d_{fi} - \tilde{d}_{fi}|/d_{fi}, . \tag{5.10}$$

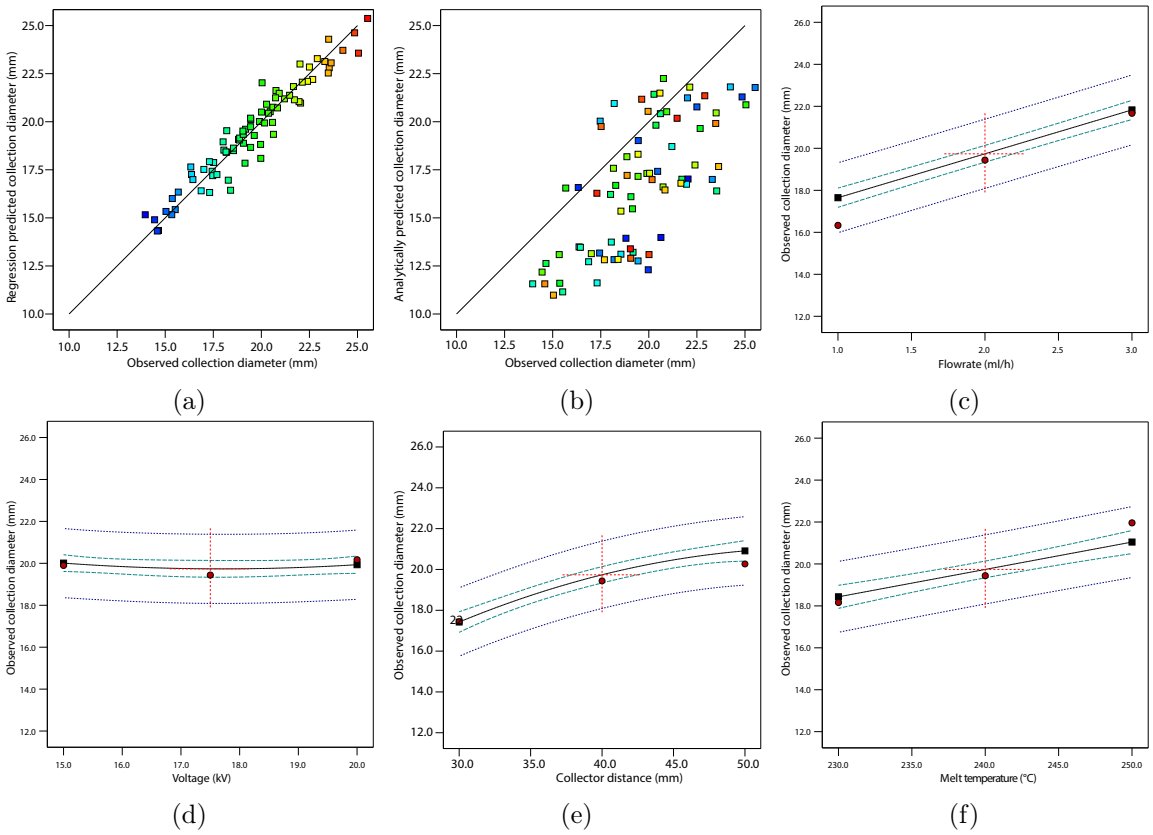


Figure 5.7: One-factor-at-a-time effects of processing parameters on **collection diameter**: (a) Regression-based prediction of collection diameters vs. observed collection diameters; (b) Analytically predicted collection diameter vs. observed collection diameter; (c) Effects of flow rate on observed collection diameter; (d) Effects of applied voltage on observed collection diameter; (e) Effects of collection distance on observed collection diameter; and (f) Effects of melting temperature on observed collection diameter.

where $\hat{\Delta}$, $\tilde{\Delta}$, and $i = 1, ..., 81$ are the regression residuals, the theoretical residuals, and the treatment order number, respectively. The subscripts c and f are short for the collection and fiber diameters, respectively.

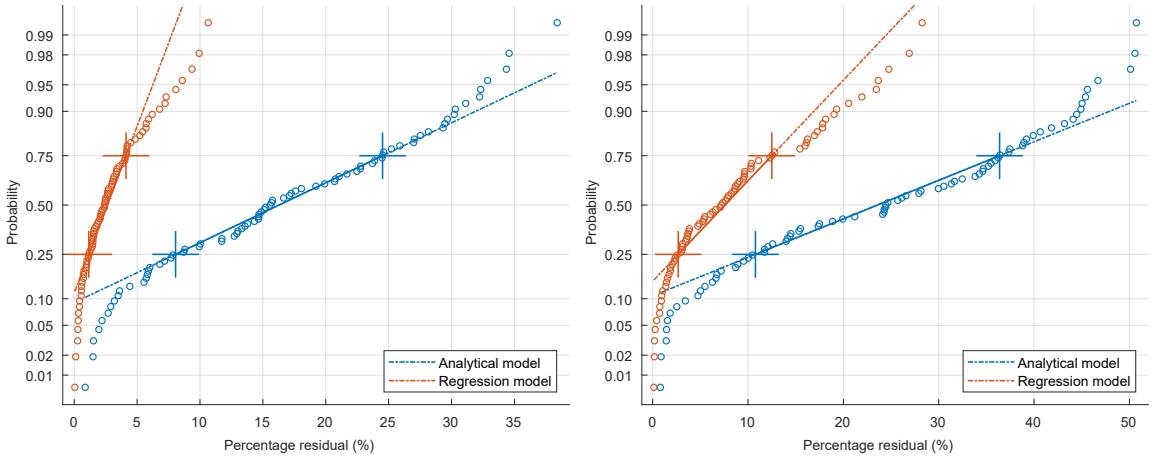
Since further statistical analysis of the residuals is valid only if the residuals are normally distributed [397], their normality is tested first in Fig. 5.8. If the residual percentage points in a given plot closely follow the line between the first and third quartiles indicated by the "+" markers, the residuals are considered normally distributed. According to this criterion, both the regression-based and the theoretically predicted collection diameter residuals in Fig. 5.8(a) were found to be normally distributed and eligible for further statistical considerations.

The coefficient of determination R^2, which could take any value between 0 and 1, is a statistical measure of how well a linear regression model fits the observation data. Values in the lower end of the range generally indicate poor fitting performance, while values close to 1 describe the best fit. The R^2 value of 0.8955 for the collection diameter regression model given by Eq. (5.5) indicates that the model has reasonable accuracy and could be safely used to navigate the design space.

A similar plot in Fig. 5.7(b) shows the theoretically predicted collection diameter values ($\tilde{d}_c$) plotted against the observed collection diameter values (d_c). Most of the data points in this plot lie below the 45° reference line, indicating that the magnitudes of the theoretically predicted collection diameters are generally less than the actual values. The probability plot of the percentage residual distribution for the theoretical model in Fig. 5.8(a) suggests a normal distribution of the residual, which is a necessary criterion for conducting further statistical analysis.

However, since the collection diameter values in the *observed vs. theoretical* plot were predicted by an analytical estimation (*i.e.*, not linear regression), using R^2 as a method of predictive performance evaluation does not apply to the theoretical model. Alternatively, Table 5.5 provides statistical assessment tools solely based on residuals the theoretical residuals. The mean (μ), standard deviation (σ), variance-to-mean ratio ($VMR = \sigma^2/\mu$), *p-value*, *critical value* (cv), and other statistical parameters are employed to evaluate the predictive performance of the theoretical model. This

approach permits drawing equivalent comparisons between the theoretical and the regression models. The fact that there is a net difference of 13.8% between the means of the two models shows a significant gap between their predictive accuracy. Stated simply, the theoretical model is generally less accurate and is susceptible to producing an average of 13.9% more error than its regression-based counterpart when predicting the collection diameter. A closer look into the VMR of the two models brings evidence of another significant performance difference. While VMR values of 1.98 and 5.75 for the regression and the theoretical models, respectively, place both models in the over-dispersion range (*i.e.*, $VMR > 1$), the theoretical model exhibits more dispersion (2.9 times more than that of the regression model). This demonstrates that besides being less accurate, the theoretical model also needs more precision when predicting the collection diameter.



(a) (b)

Figure 5.8: Plots of normal distribution of residuals: (a) Collection diameter; (b) Average fiber diameter

Table 5.5: Regression vs. theoretical model comparisons based on collection diameter prediction

Collection diameter (mm)	μ	σ	**95% CI** (μ)	**95% CI** (σ)	$p-value$	ks	cv
Regression formula	2.9	2.4	[2.4, 3.5]	[2.1, 2.8]	0.0011	0.1351	0.1163
Theoretical model	16.7	9.8	[14.5, 18.9]	[8.5,11.6]	0.3951	0.0712	0.0998

Table 5.6: Regression vs. theoretical model comparisons based on fiber diameter prediction

Fiber diameter (µm)	μ	σ	**95% CI** (μ)	**95% CI** (σ)	$p-value$	ks	cv
Regression formula	8.5	7.2	[6.9, 10.2]	[6.2, 8.6]	0.0035	0.1259	0.1163
Theoretical model	24.2	14.9	[20.8, 27.5]	[12.9, 17.7]	0.0516	0.0995	0.0998

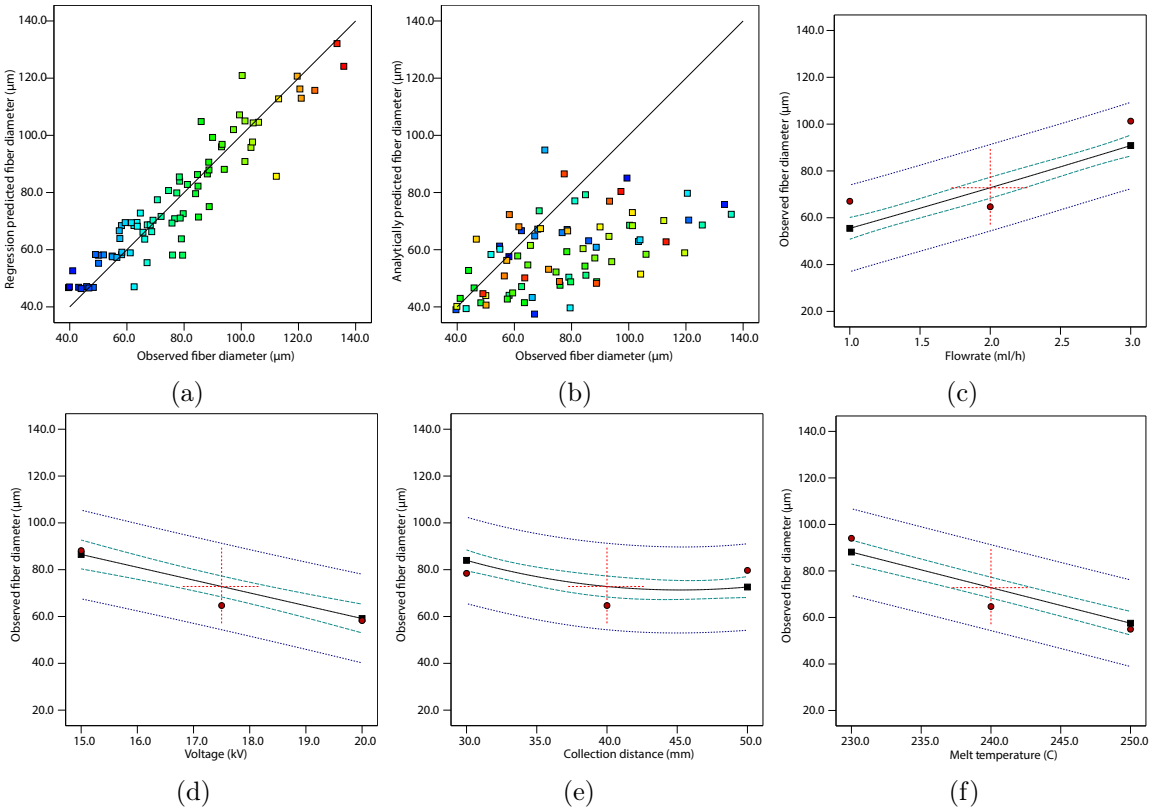


Figure 5.9: One-factor-at-a-time effects of processing parameters on **fiber diameter**: (a) Regression-based prediction of fiber diameters vs. observed fiber diameters; (b) Analytically predicted fiber diameter vs. observed fiber diameter; (c) Effects of flow rate on observed fiber diameter; (d) Effects of applied voltage on observed fiber diameter; (e) Effects of collection distance on observed fiber diameter; (f) Effects of melting temperature on observed fiber diameter.

5.4.4 Average Fiber Diameter Prediction Performance

Similar plots in Figs. 5.9(a) and 5.9(b) show comparisons of the average fiber diameters predicted by the regression model and the theoretical model, respectively, with the corresponding experimentally observed fiber diameters.

Comparisons between the regression and the theoretical model's average fiber diameter predictions follow similar trends to the collection diameter. While the residuals of the regression model are dispersed evenly on both sides of the 45° reference line, most of the theoretically predicted values are concentrated slightly below the reference line. This shows that the theoretically predicted average fiber diameters are usually slightly smaller than the actual values. The R^2 value of 0.8232 for the regression model is still reasonably acceptable. However, a closer look into the means

of the percentage residuals of the fiber diameter prediction reveals a significant drop in the accuracy of both the regression and the theoretical models compared to the collection diameter predictions. Specifically, the predictions are more likely to be out by 8.5% and 24.2% for the regression and the theoretical models, respectively. The variance-to-mean ratio of 6.1 and 9.2 for the regression and theoretical models, respectively, also highlight significant deterioration in the dispersal behaviors of both models compared to their collection diameter counterparts.

Generally, both the regression and the theoretical models performed with better accuracy and precision when estimating the collection diameters than the average fiber diameter. Because they were derived from the observation data, the regression models are expected to perform better than the theoretical model both in the collection and the average fiber diameter predictions. Of course, this is the primary reason they are included in the current work in the first place; to serve as benchmarks against which the performances of the theoretical models are objectively and quantitatively assessed.

5.4.5 Factors contributing to the loss of prediction accuracy

Overall, relative to the regression models, the theoretical model performed inferiorly in predicting both the collation and average fiber diameters. Several factors may explain this. First, the single-layer viscoelastic conceptualization in the theoretical model (Fig. 5.2) is still too simplistic and inadequate to fully capture the material property. It does not have the necessary degrees of freedom (resolution) to properly curve-fit the rheological data. As demonstrated in Table 5.2, Fig. 5.5(d) and 5.5(e), at least 10–15 layers of Maxwell links were required to properly curve-fit the viscoelastic behavior of the material with a reasonable accuracy [398–401]. However, employing such rheological models in the MES modeling brings its challenges. The resulting equations of motion would involve higher-order partial time derivative terms of the stress variable $\sigma(t)$ up to the order of $N+1$ (*i.e.*, $\frac{\partial \sigma}{\partial t}$, $\frac{\partial^2 \sigma}{\partial t^2}$, ..., $\frac{\partial^{(N+1)} \sigma}{\partial t^{(N+1)}}$). Numerically integrating such problems requires estimating initial values for all orders of the stress derivative terms, which is an arduous task, if not impossible. Another disadvantage of having a higher-order stress derivative term is that the problem becomes computa-

tionally expensive to solve. These are the primary reasons the generalized viscoelastic model in the theoretical model is limited to having only one layer.

The other factor affecting the theoretical model's prediction accuracy is the material's temperature profile inside the printing chamber. The temperature of the molten material drops as it leaves the nozzle and traverses through the air toward the collector plate. This temperature change inversely affects the elastic parameters of the material. The elastic modulus (E_i) and viscous damping coefficients (η_i) are expected to rise due to the temperature drop, reducing the rate of fiber thinning as the jet travels further away from the nozzle tip. However, the exact magnitude and rate of drop are unknown and need to be appropriately investigated. The temperature of the fiber is assumed to remain unchanged once it leaves the nozzle to simplify the modeling process in the current work, regardless of the lower ambient temperature inside the chamber. Maintaining a higher melting temperature throughout the chamber means preserving the initial lower elastic modulus and viscosity, which promote further thinning of the fiber as brought to evidence by the simulation results (Fig. 5.9(b)). However, this is not the case in actual MES operations, where the fiber develops resistance to further elongation once it leaves the nozzle in response to a sudden drop in ambient temperatures.

Torsional elastic and viscous damping elements were introduced in the theoretical model to capture the viscoelastic properties of the fiber in the rotational (bending) sense (Fig. 5.1). Without these, the energy dissipation would occur only for deformations along the longitudinal axis of the fiber, potentially allowing the fiber to vibrate laterally indefinitely without any dampening in the whipping motion. Although the theoretical model was developed with provisions to simulate the effects of these viscoelastic elements included, the parameter values used in the simulations were only roughly estimated as ratios of their linear counterparts due to time and scope limitations. Further work is needed to develop this fully.

Considering the simplifying assumptions made and the gaps and challenges identified in material characterizations, the authors concur that the theoretical predictive model has performed reasonably. The authors believe that, with these challenges

addressed, the accuracy and precision of the prediction of the theoretical model could be improved significantly. With the current mean percentage-residual values of 16.7% for the collection diameter and 24.2% for the fiber diameter, the theoretical model prediction capabilities are still considered reasonably acceptable and encouraging.

5.5 Conclusion

A theoretical predictive model for melt electrospinning, which was previously developed, was briefly discussed at the beginning to provide the reader with some background information. The current work employed this theoretical model to predict the experimental collection and average fiber diameter values under three discrete levels of four processing parameters: flow rate, collector distance, applied voltage, and melt temperatures. Experiments were carried out to observe the outputs of the melt electrospinning process under similar processing conditions. The experimental data was later used to generate two predictive regression models—one for predicting the collection diameter and another for the fiber diameter. The outputs of the theoretical and the regression models were compared to evaluate the prediction performance of the theoretical model. A discussion has been presented on the possible sources of errors in the theoretical model. Generally, challenges associated with material characterization under varying ambient temperatures and challenges in numerical solving of higher-order differential equations were identified as the main contributing factors. According to the study, both the collection diameter and fiber diameter theoretical predictive models performed reasonably accurately, given the challenges and gaps identified.

Chapter 6

Verification of Direct Writing Capabilities and Future Directions

6.1 Introduction

Direct writing in MES involves the deposition of the melt-electrospun fibers in a predefined pattern. This process has gained significant research attention recently as it is the first critical step toward printing three-dimensional objects with a sizable thickness along the z-axis direction [160, 321, 402–404]. Unlike the traditional fused deposition modeling (FDM) printing techniques, melt-electrospun fibers are not extruded from the nozzle. Instead, it is drawn under the actions of electrostatic forces from its molten state. The process is usually characterized by erratic whipping motions of the jet stream, which makes it too difficult to control in the traditional sense of 3D printing.

The jet usually starts with a stable and straight flow pattern near the nozzle tip. As it travels away from the nozzle and gets closer to the collector plate, it behaves in an uncontrolled manner with a random-looking whipping motion. The further the jet from the nozzle, the higher the whipping magnitude. The experimental investigation in Chapter 5 has shown that the collector diameter, a direct measure of the whipping magnitude, is inversely proportional to the collector distance. The further the collector distance, the more prominent the magnitude and randomness of the accumulation become, resulting in a poor boundary definition in the deposited

fiber. On the other hand, if the collector plate is kept at a reasonably closer distance to the nozzle, equivalent to capturing the accumulation at the stable region, it would result in a narrow and well-defined deposition pattern [405–407]. All intermediate distances could be tapped to achieve varying definitions.

Collector distance is not the only factor that dictates the characteristics of the accumulated fiber. The melt temperature, applied voltage, and flow rate all lend some degree of leverage that could be utilized to control the outcome [408, 409]. The melt temperature, for example, is the primary factor responsible for the state of the material's elastic properties, which in turn dictates how easily the fibers draw under a given set of processing conditions. On the other hand, the applied voltage influences both the available electric field force and the inter-particle electrostatic charge distribution, dictating the degree of fiber stretching again [410]. Increasing the applied voltage increases the electric field and the Coulomb forces, yielding relatively thinner fiber formations. Flow rate also dictates how much material is available for drawing. The higher the flow rate, the thicker the fiber, and vice versa.

Melt-electrospinning might not be the ideal fabrication method if the goal is to achieve smooth profiles at the boundaries of the accumulations. However, this same behavior may be a key advantage for applications requiring some degree of surface roughness and randomness in accumulations. One such application area is bone tissue engineering, where the internal structure of the scaffold is required to have some roughness and porosity for cell attachment, seeding, proliferation, and metabolic activities. Achieving such micro-architectural details with traditional fabrication methods could be challenging as the geometrical details would have to be modeled in advance. In addition, only a few free-form manufacturing systems are capable of transforming such detailed designs into actual constructs.

Melt-electrospinning has excellent potential to address some of these challenges. It has already been proven to produce accumulations that functionally resemble the extra-cellular matrix (ECM) of the natural bone. Further, tailored deposition outcomes could be realized by controlling the processing parameters. As discussed in Chapter 2, several experimental research efforts have been on this subject, particu-

larly to study the effects of process parameters on fiber characteristics. The current study aims to complement these efforts with a theoretical approach. In addition to being a valuable tool for estimating important fiber attributes, such as average fiber and collection diameters, the proposed theoretical model could also be used to simulate direct writing scenarios, a precursor to printing three-dimensional objects. The upcoming sections are demonstrations of these capabilities.

6.2 General Direct Writing Workflow

In this chapter, the direct writing capabilities of the theoretical predictive model are demonstrated. The process typically starts with digitally creating the target image using third-party CAD or vector graphic design software. The image is next processed to generate paths the nozzle must follow during writing. Once the paths are created, they are exported in a G-code format, a structured text language that most CNC machines and 3D printers require to run. These first three steps are usually handled using different software. However, some software packages also provide integrated environments where everything from geometrical modeling to G-code generation can be done within the same application. The generated G-code is next sent to the MES 3D printer developed for this purpose. The predictive simulation software also is given the same G-code as one of its primary inputs. After running the printer and the simulation model using the same data, the results are compared and analyzed to study the fiber accumulation characteristics. Fig. 6.1 summarizes this general workflow.

6.3 Demonstration of Direct Writing Capabilities

6.3.1 Example 1: Writing an Image

As a first demonstrative example, the "MFCL@UofA" image shown in Fig. 6.2, which is an acronym for the *Multi-functional Composite Laboratory at the University of Alberta*, where this study was conducted, was processed as per the workflow in Fig. 6.1. The image was first created in Inkscape®, an open-source vector graphics editing software. This tool was selected for this task because of its open-source accessibility

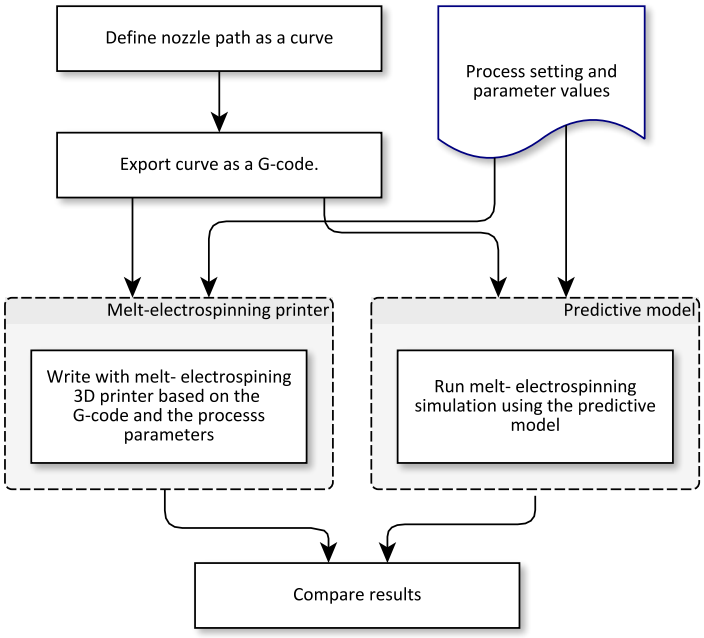


Figure 6.1: Direct writing process workflow

and ability to generate G-codes. The image was created in a one-to-one ratio to the size of the collector plate platform on the custom printer (160 mm × 140 mm).

A G-code (Fig. 6.3) was generated from the image using a 500 mm/min feed rate. The translational speeds of extruders and tools in 3D printers and CNC machines are usually expressed in mm/min units. The same convention is followed here to take advantage of existing software and controllers. The second example in the upcoming section discusses the effects of different feed rates on the deposition behavior of the fibers.

The MES 3D printer runs directly from the G-code without needing additional modifications or code editing. However, the simulation software requires intermediate steps to translate the G-code to the nozzle kinematic profile in a manner compatible with the dynamic problem formulation discussed in Chapter 4. To achieve this, a custom G-code interpreter function was written in MATLAB®that imports and interprets the G-code file. This function reconstructs the nozzle path and calculates appropriate position, velocity, and acceleration values for all numerical time steps.

As discussed in Chapter 4, the nozzle kinematics are parametrically included in

Figure 6.2: Image to be processed for simulation and actual direct writing

```
%
(Header)
(Generated by gcodetools from Inkscape.)
(Using default header. To add your own header create file "he
M3
(Header end.)
G21 (All units in mm)

(Start cutting path id: path2541)
(Change tool to Default tool)

G00 Z29.000000
G00 X134.753199 Y71.120611

G01 Z30.000000 F100.0(Penetrate)
G01 X136.152898 Y71.120611 Z30.000000 F500.000000
G01 X147.245957 Y47.099826 Z30.000000
G03 X148.199635 Y46.088762 Z30.000000 I1.973632 J0.906306
G03 X149.982666 Y45.644021 Z30.000000 I1.783030 J3.351847
G01 X150.943654 Y45.644021 Z30.000000
G01 X150.943654 Y44.312493 Z30.000000
G01 X138.618015 Y44.312493 Z30.000000
G01 X138.618015 Y45.644021 Z30.000000
G01 X139.599896 Y45.644021 Z30.000000
G03 X141.495514 Y45.767416 Z30.000000 I-0.000000 J14.622217
G03 X142.315712 Y45.999095 Z30.000000 I-0.381583 J2.918576
G03 X142.722398 Y46.335281 Z30.000000 I-0.424438 J0.927524
G03 X142.858873 Y46.762507 Z30.000000 I-0.600466 J0.427226
G03 X142.825566 Y47.057407 Z30.000000 I-1.322175 J0.000000
```

Figure 6.3: G-code generated from the target image

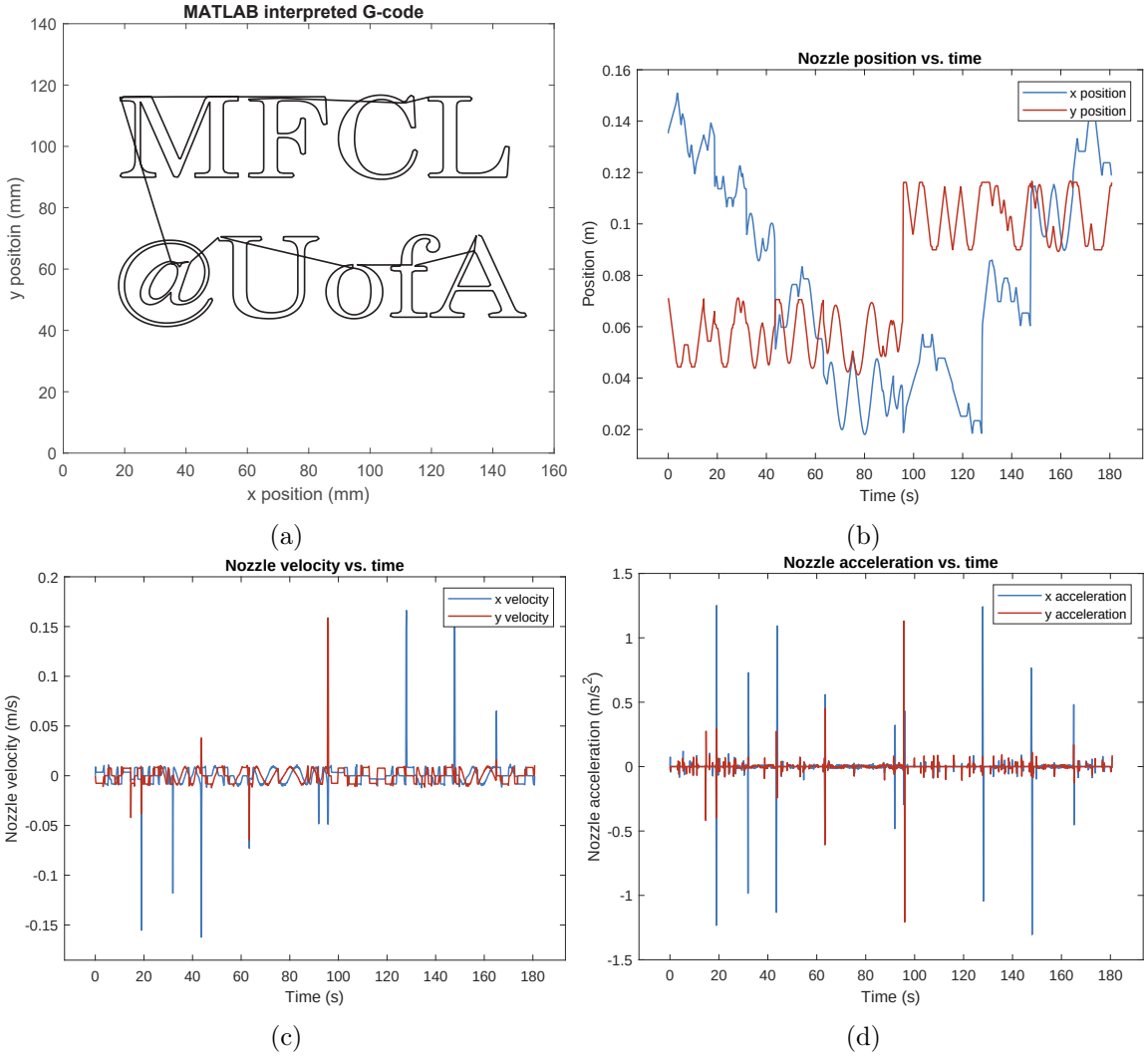


Figure 6.4: Kinematic profiles of the nozzle path: (a) Target image reconstructed from G-code in MATLAB®; (b) Nozzle x and y positions as functions of time; (c) The velocity profiles of the nozzle; (d) Nozzle acceleration profiles

the dynamic formulation as constraint equations. The constraint parameters were extracted from the G-code. Each axis motor on the 3D printer ramps up and down to the feed rate speeds in response to commands from the G-code. Although the actual motor accelerations and decelerations are the results of both the commanded signal and the internal inertial properties of the motors, the MATLAB®interpreter function ignores any contribution made by the latter. Hence, the motors are assumed to respond to the G-code commands without delay. The velocity and acceleration profiles shown in Fig. 6.4(c) and (d) were therefore simple time derivative of the position of the nozzle in Fig. 6.4(b).

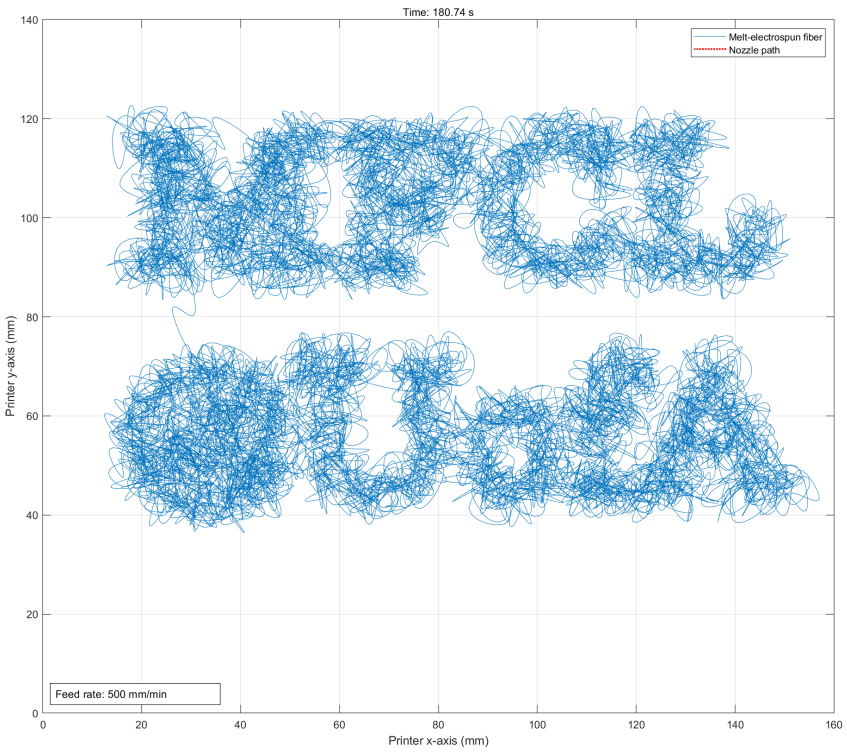
The simulation and the actual direct writing were run based on the following processing parameters: collector distance = 30 mm; melt temperature = 250 °C; applied voltage = 15 kV; flow rate = 2 mL h^{-1}; and ambient temperature of 60 °C. Fig. 6.5(a) and (b) show the final simulated and actual direct writing results, respectively.

6.3.2 Example 2: Investigation of Effects of Nozzle Speed on Fiber Characteristics

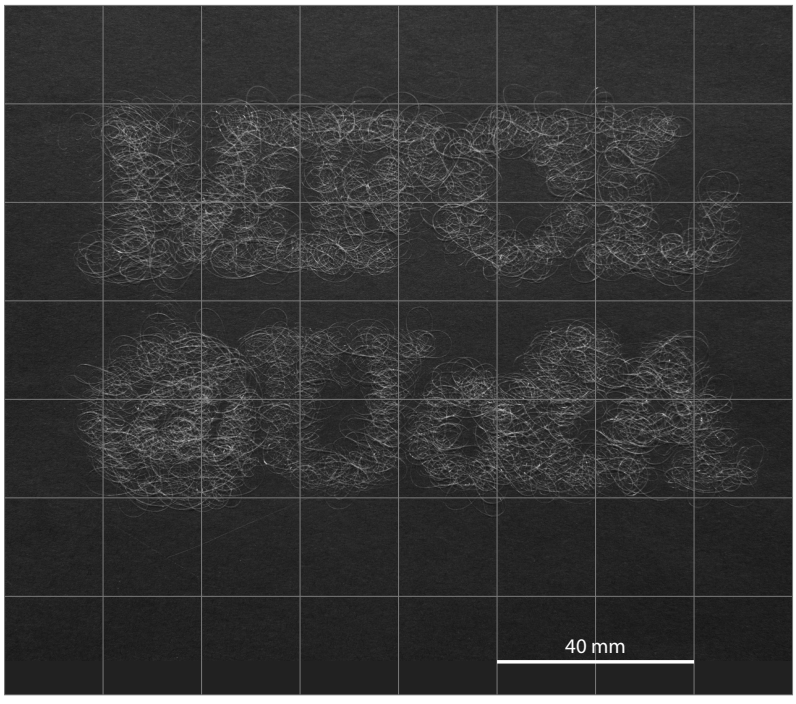
As a second demonstration, the entanglement characteristics of fiber accumulations under different translational nozzle velocities (feed rates) were investigated. For this example, a simple rectangular path of 80 mm × 60 mm dimension was created in Inkscape®and exported to G-code with 500 , 2000 , 4000 , 10 000 , 50 000 , and 100 000 mm/min feed rates. Although the maximum feed rate on the printer was limited to 10 000 mm/min, the additional two feed rates were included in the simulation to investigate the effects of extreme speeds.

The simulation results in Fig. 6.6(a)–(d) are in total agreement with the actual observations in (Fig. 6.7(a)–(d). Both indicate that the fiber entanglements open up as the feed rate increases, leading to a stretched line deposition pattern. It is notable from the simulation at higher speeds (Fig. 6.6(e)–(f)), the fibers accumulation at the corners show signs of momentum imparted by the moving nozzle.

In both examples above, the simulated results and the actual-direct written accumulations showed agreement on the macro level. They displayed similar accumulation



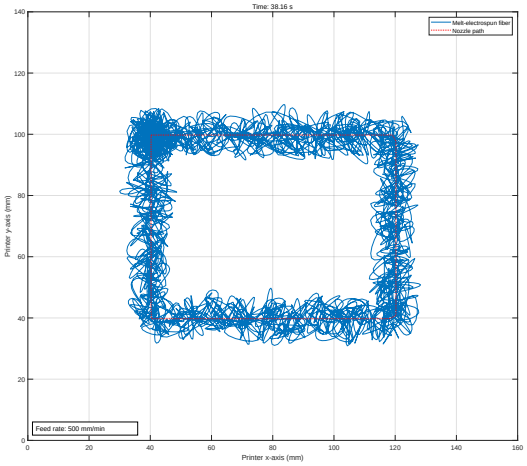
(a)



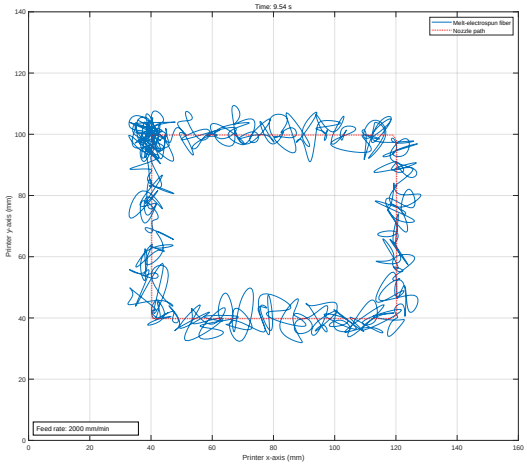
(b)

Figure 6.5: Simulated and direct-written images: (a) Simulation output; (b) Melt-electrospun image based on G-code

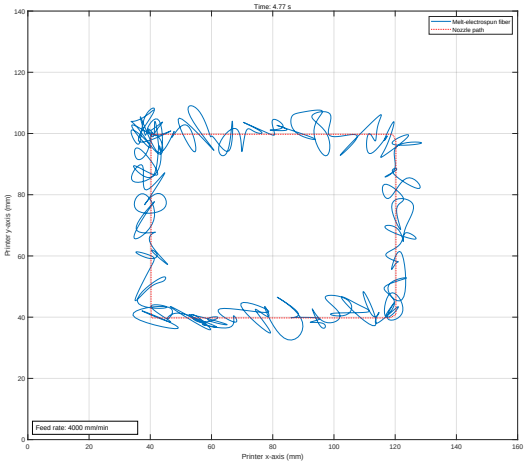
(a) Feed rate: 500 mm/min



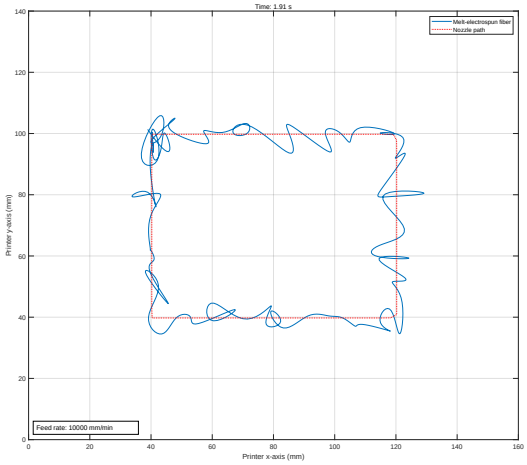
(b) Feed rate: 2,000 mm/min



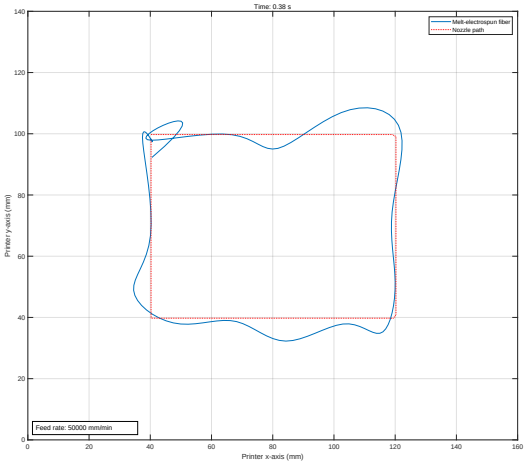
(c) Feed rate: 4,000 mm/min



(d) Feed rate: 10,000 mm/min



(e) Feed rate: 50,000 mm/min



(f) Feed rate: 100,000 mm/min

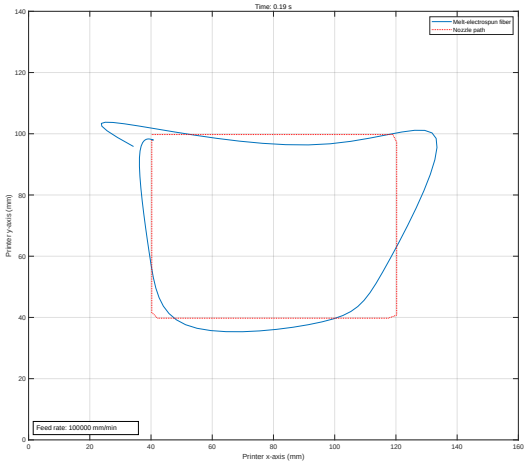


Figure 6.6: Simulated effects of lateral nozzle motion on fiber deposition characteristics

(a) Feed rate: 500 mm/min

(b) Feed rate: 2,000 mm/min

(c) Feed rate: 4,000 mm/min

(d) Feed rate: 10,000 mm/min

Figure 6.7: Effects of direct writing at different feed rates

patterns, densities, and boundary definitions, supporting our initial argument that the predictive model could be a valuable tool to simulate direct writing applications.

There were, however, some subtle differences between the simulated and the actual fiber accumulations. For example, a closer visual inspection of the shapes of each strand in the two outputs showed different accumulation patterns. Although several factors could explain this observation, the nature of the multi-physics problem itself is believed to be the main contributor. From the point of view of dynamical systems, the mathematical model is highly chaotic and is sensitive to both the initial conditions and process parameter values, whose estimation methods still need to be perfected. In addition, several parameter values must be appropriately estimated under varying processing and environmental conditions. Given the chaotic nature and the size of each fiber model (length of the serially-connected multi-body system), there is doubt if the initial values and estimated parameters used in the simulation will ever match their real-world counterparts. For this reason, our focus has been on using the model in estimating high-level accumulation attributes that have practical implications in bone scaffold applications. These include the prediction of the general pattern and shape of the construct, the collection density, and the average fiber diameter, for which the model's predictive accuracy has been proven reasonably acceptable.

6.4 Future Directions—Thickness Building

The next logical step for this study is to develop methodologies for controlling variable deposition densities in the two-dimensional sense. Algorithms to translate deposition density specifications into equivalent nozzle paths may be required. Such a study will be vital as it is a critical step toward understanding factors dictating the internal micro-architectures of melt-electrospun bone scaffolds.

Building three-dimensional constructs with spatially distributed topological properties is another area that needs to be investigated. Thickness building requires improving both the prediction model and the MES printing equipment. For one, the effects of growing material deposition must be understood and theoretically modeled. The current model does not come with this feature. The prediction is limited only to

the first several layers of accumulations where the electric field is assumed unaffected by the growing material deposition.

Building thickness also involves automatically and dynamically adjusting the process parameters, such as the collector distance or the applied voltage, to compensate for the effects of changes in other variables. The interplay among these parameters needs to be understood and accurately modeled. The G-code fed to the printing machine must also be enhanced to incorporate this additional information. The printing equipment would need a controller and firmware that reads and interprets this new information set and automatically adjusts the process parameters for the different layers. Variable high-voltage power supply units, controllable by G-codes, may be required.

Chapter 7

Conclusion and Future Work

This chapter contains two main sections. The first section is the Conclusion, which gives a short overview of the research problem and methodology, along with the key findings of the study and its uniqueness. The second section discusses the limitations of the proposed model and recommends areas for future development, offering valuable insights that could enhance the model's effectiveness in practical applications.

7.1 Conclusion

The main objective of this research was to develop a forward predictive model for simulating MES and estimating fiber characteristics given a specific set of material and process parameters.

Building on previous works by other groups, a more advanced theoretical model was formulated in the current study by conceptualizing fibers as serially-connected multi-body systems consisting of smaller fiber segments called *beads*. The proposed model included improvements such as using Maxwell's generalized viscoelastic rheological model as a connective element between successive beads in the multi-body system. Moreover, for the first time, the dynamic formulation included the effects of the nozzle and the collector plate motions, which are common in most 3D printing applications. The anchoring effect of the fiber sticking on the collector plate was also included in the problem formulation in the form of a set of constraint equations.

Kane's Dynamical Equations and The Udwadia-Kalaba methods were employed to formulate the unconstrained equations of motion and handle constraint equations, respectively. To this end, a total of seven independent generalized coordinates were defined per bead: three as components of a position vector and the remaining four as quaternion parameters for orientation representation.

An algorithm was also proposed that allows continuous simulation of the MES process without the need to define the size of the problem ahead of time. Unlike most multi-body systems, the MES conceptualization in the current study involved the ejection of new beads (representing new materials) from the nozzle at regular intervals and the retirement of older beads when the fiber gets deposited at the collector plate. The proposed algorithm allowed this continuity by dynamically resizing the problem to match only the number of active beads (active fiber segment between the nozzle and the collector plate). The advantage of this approach is its efficient utilization of computational resources and the flexibility to simulate the process indefinitely. The model was implemented in MATLAB®. As a result of this new development, it was possible to simulate and analyze the effects of different translational speeds of the nozzle on the behavior of the fiber depositions.

A design of experiments (DOE) was carried out to validate the predictive model and the accuracy of its implementation. A custom MES equipment was developed by modifying a commercially available 3D printer with a high-voltage supply, electrical insulation, and chamber heater and temperature controller. A custom firmware was also created for melt-flow rate control.

A total of 81 simulations and experimental runs were carried out based on the designed experiment with different treatments of the controlled processing parameters (applied voltage, collector distance, flow rate, and melt temperature). A comparison of results generated from the two showed that the predictive model has reasonable accuracy in predicting collection and average fiber diameters (16% and 24% average errors, respectively). The experimental data were also fitted to generate regression-based predictive models for the collector and average fiber diameters. However, since the data was collected using PLA material and a vertical spinning configuration, the

applicability of models is restricted to only matching cases.

Although it is novel and advanced in many ways compared to the existing models, the proposed model is still a forward prediction tool, far from the reverse predictive capabilities sought for optimizing the process parameter values to achieve targeted fiber characteristics. Much work needs to be done to develop the proposed model to enable process optimization and reverse predictive capabilities.

Using a Maxwell standard linear solid model (SLS) in the proposed model has significantly improved the accuracy of the viscoelastic properties of the materials. The additional link in the rheological model has added two more degrees of freedom to the rheological model, which enhanced the curve-fitting performance. From this perspective, a large number of layers in the model is a desirable attribute. However, some practical challenges have limited the maximum number of layers used in the proposed predictive model. One of these challenges stemmed from difficulties in estimating initial values for higher-order stress and strain derivative terms in the resulting equations of motion. If there are N number of layers in the system, for example, the resulting equations would have time derivative terms of the stress and strain variables up to the N^{th} order. Estimating initial values for these higher-order variables would be challenging as no physical meanings could be attributed to the derivative terms. In addition, as a chaotic system, a slight variation in estimating the initial values could also have led to significantly divergent simulation outcomes. It was necessary to strike a balance between the advantages of having precise material representations obtained from utilizing multiple layers and the possible errors that could arise from inaccurately estimated initial values. The proposed model suffered from this dilemma. A choice was made to adopt fewer layers in the model to avoid the difficult task of estimating initial conditions for non-existent physical quantities. This decision also had the added benefit of having a computationally lightweight and efficient model but came at the cost of losing predictive accuracy.

The study was concluded by demonstrating the capabilities of the predictive model in simulating direct-writing applications where the fibers were deposited in a predefined pattern. As demonstrative examples, two pattern geometries were designed

using third-party software and exported into G-code at different translational nozzle velocities (*feed rates*). These G-codes were used as inputs both in the custom MES printer and the simulation software. Even though several areas could use improvement, the current results were found to be reasonably acceptable. The effect of nozzle feed rate on fiber entanglement topology generated by the simulation was in agreement with the actual outcomes from the experiments.

7.2 Limitations, Recommendations, and Future Work

This section addresses the current model's limitations and suggests areas that require further development.

7.2.1 Parameter Estimation

Simulation of the proposed model also relies on several other parameter values. Every effort has been made to collect most of these values from experiments. However, some parameter values could not be determined from experiments promptly due to time constraints. These include coefficients used in numerical stabilization, the dielectric constant of the material, and elastic material parameters used in the rotational sense. As demonstrated in Chapter 5, the predictive model is already performing with reasonable accuracy based on rough estimates of these parameter values. While some parameters are experimentally calculated, others were collected from datasheets and the literature. The predictive accuracy could be improved by refining the values using parameter estimation techniques. One example of such parameters is the surface charge density of the material. The current model relies on a custom function that returns a static value. This function needs to be updated to return values based on the dynamic processing conditions affecting the parameter, such as the applied voltage, the collector distance, and the melt temperature. Other parameters needing similar improvements may include the torsional elastic parameters k_t and c_t.

All in all, about ten parameters need to be appropriately estimated. The current model's parameter estimation functions are designed in a modular way. This approach allows for future improvements to be made independently, without affecting the model's structure, when time and resources permit.

7.2.2 Material Characterization

The two-layer generalized Maxwell viscoelastic rheological model adapted in the model is too simplistic to capture the material characteristics fully. Additional layers may be required for better curve fitting. However, the additional layers introduce challenges in estimating initial conditions for higher-order derivative terms of the stress and strain variables. More work is needed to address this challenge and develop a practical methodology.

7.2.3 Elastic Parameter Profiles Under Dynamically Varying Temperature Conditions

Oversimplified spatially and dynamically varying temperature distributions inside the printing chamber may lead to incorrect simulation results in the current model.

If steady-state conditions with no air movement inside the printing chamber are assumed, there would be a static temperature gradient between the nozzle and the chamber temperature. This gradient is expected as long as the nozzle and the chamber are set to different temperatures. Under these circumstances, the temperature readings at various points inside the chamber would be different, but the reading at a given point would remain the same throughout the process. The spatial temperature distribution, in this case, could be written as a function of the position coordinates, *i.e.*, $T(x, y, z)$.

Under the above circumstances, when the material travels from one region to another, it passes through different temperature zones. How fast the elastic properties of the material change in response to variations in temperature has yet to be thoroughly understood. Specifically, it is not clear at what rate the material's elastic constants E_0 and E_1 and viscosity (η) increase in response to drops in temperature. If the effect of the temperature change on the elastic parameters does not take effect immediately, the jet will maintain its initial elastic properties. The result would be extreme fiber elongations and randomness in the accumulation pattern. On the other hand, if the temperature response is relatively fast and happens before the fiber is accumulated on the collector plate, changes in the material properties would be observed. These

changes may lead to rapid fiber solidification right after elongation.

The latter scenario is what the author believes to be occurring during most MES operations. Although exposed to low ambient temperature for a brief period, the surface area of the fiber significantly increases due to the initial stages of fiber drawing. The increased surface area creates an ideal condition for high heat transfer rates that ultimately lead to the solidification of the fibers. Once in a solid state with a relatively higher bending resistance than the jet-initiating stages, the fibers tend to buckle, forming coils. The fiber's mechanical strength starts dominating the fiber topology as the fibers cool down near the collector plate. This observation underscores the fact that the elastic parameters of the material are not only temperature-dependent but also time-dependent (*i.e.*, $E_0(T(x,y,z),t)$, $E_1(T(x,y,z),t)$, $\eta(T(x,y,z),t)$).

More work needs to be done to model these parameters as functions of both temperatures, by extension positions, and time. The elastic parameter estimation function in the current model needs to be updated to account for the dynamic responses to temperature changes.

7.2.4 Dynamic Interplay Between Processing Parameters under Increasing Build Thickness

Changing the collector distance alters the electric field intensity between the nozzle and the collector plate. As demonstrated experimentally in Chapter 5, the average fiber diameter and the collection diameter are affected by changes in collector distance. Suppose uniform morphological properties are required along the z-axis (thickness) in the 3D-printed construct. Suitable adjustments in the other processing parameters, such as the applied voltage, must be made to compensate for the effects caused by changing collector distance. This study has theoretically and experimentally identified the interdependencies among the processing parameters and their compounded effects on fiber topologies. However, a suitable strategy for compensating the values of the processing parameters has yet to be proposed.

7.2.5 Effect of Growing Material Accumulation on Electric Field

One area of uncertainty is the effect of accumulated material on the electric field properties. Does a significant amount of fiber accumulation between the nozzle and the collector plate interfere with the electric field? Simple experiments have shown that growing fiber accumulations at stationary points appear to deflect incoming fiber streams away from the center of accumulation. An in-depth experimental investigation and further theoretical modeling may be required in this area.

7.2.6 The Effect of Build-sheet Material on Electric Field Characteristics

Collecting the fibers on different substrate materials, such as aluminum foil, paper, or plastic sheet, has led to different topological outcomes. While the smallest collection diameter was observed on the aluminum foil, the plastic sheet resulted in the largest diameter. It is essential to comprehend the mechanism behind this phenomenon fully.

7.2.7 Modeling of Spatial Distribution of Electric Field

In the current study, the electric field between the nozzle and the collector plate was approximated by a uniform distribution as if it was generated between two parallel plates. This simplifying assumption is believed to be one of the error contribution factors. The electric field model could be updated to reflect the actual geometry and material of the electrodes (the nozzle and the collector plate).

With these improvements, the prediction model's accuracy could be increased. The model could be vital in accelerating the application of MES in bone tissue engineering applications.

Bibliography

[1] S. S. D. Carter, P. F. Costa, C. Vaquette, S. Ivanovski, D. W. Hutmacher, and J. Malda, "Additive Biomanufacturing: An Advanced Approach for Periodontal Tissue Regeneration," *Annals of Biomedical Engineering*, vol. 45, no. 1, pp. 1–11, 2016. DOI: 10.1007/s10439-016-1687-2.

[2] A. Wubneh, E. Tsekoura, C. Ayranci, and H. Uludağ, "Current state of fabrication technologies and materials for bone tissue engineering," *Acta Biomaterialia*, vol. 80, 2018. DOI: 10.1016/j.actbio.2018.09.031.

[3] B. Mirani, S. Mathew, N. Latifi, M. Labrosse, B. Amsden, and C. Simmons, "Programmable melt electrowriting to engineer soft connective tissues with prescribed, biomimetic biaxial mechanical properties," Jul. 2022. DOI: 10.21203/rs.3.rs-1851887/v1. [Online]. Available: https://doi.org/10.21203/rs.3.rs-1851887/v1.

[4] S. Becker *et al.*, "Biocompatibility of individually designed scaffolds with human periosteum for use in tissue engineering," *Journal of materials science. Materials in medicine*, vol. 21, pp. 1255–62, Feb. 2010. DOI: 10.1007/s10856-009-3878-y.

[5] P. Lichte, H. Pape, T. Pufe, P. Kobbe, and H. Fischer, "Scaffolds for bone healing: Concepts, materials and evidence," *Injury*, vol. 42, no. 6, pp. 569–573, 2011, Bone Regeneration in the 21st Century. DOI: https://doi.org/10.1016/j.injury.2011.03.033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020138311001276.

[6] N. A. Sears, D. R. Seshadri, P. S. Dhavalikar, and E. Cosgriff-Hernandez, "A review of three-dimensional printing in tissue engineering," *Tissue Engineering Part B: Reviews*, vol. 22, no. 4, pp. 298–310, 2016, PMID: 26857350. DOI: 10.1089/ten.teb.2015.0464. eprint: https://doi.org/10.1089/ten.teb.2015.0464. [Online]. Available: https://doi.org/10.1089/ten.teb.2015.0464.

[7] F. Torres, S. Nazhat, S. Fadzullah, V. Maquet, and A. Boccaccini, "Mechanical properties and bioactivity of porous plga/tio2 nanoparticle-filled composites for tissue engineering scaffolds," *Composites Science and Technology*, vol. 67, pp. 1139–1147, May 2007. DOI: 10.1016/j.compscitech.2006.05.018.

[8] V. S. Venkataiah *et al.*, "Clinical applications of cell-scaffold constructs for bone regeneration therapy," *Cells*, vol. 10, no. 10, p. 2687, Oct. 2021. DOI: 10.3390/cells10102687. [Online]. Available: https://doi.org/10.3390/cells10102687.

[9] J.-H. Zeng *et al.*, "Scaffolds for the repair of bone defects in clinical studies: A systematic review," *Journal of Orthopaedic Surgery and Research*, vol. 13, no. 1, Feb. 2018. DOI: 10.1186/s13018-018-0724-2. [Online]. Available: https://doi.org/10.1186/s13018-018-0724-2.

[10] F. Zhang, K. Cao, A. Zaeri, R. Zgeib, and R. C. Chang, "Design, fabrication, and analysis of spatially heterogeneous scaffold by melt electrospinning writing of poly(ϵ-caprolactone)," *Journal of Applied Polymer Science*, vol. 139, no. 22, p. 52 235, Feb. 2022. DOI: 10.1002/app.52235. [Online]. Available: https://doi.org/10.1002/app.52235.

[11] J. White, M. Foley, and A. Rowley, "A novel approach to 3d-printed fabrics and garments," *3D Printing and Additive Manufacturing*, vol. 2, pp. 145–149, Sep. 2015. DOI: 10.1089/3dp.2015.0019.

[12] J. Fang, L. Zhang, D. Sutton, X. Wang, and T. Lin, "Needleless melt-electrospinning of polypropylene nanofibres," *Journal of Nanomaterials*, vol. 2012, Jun. 2012. DOI: 10.1155/2012/382639.

[13] D. Hutmacher and P. Dalton, "Melt electrospinning," *Chemistry, an Asian journal*, vol. 6, pp. 44–56, Jan. 2011. DOI: 10.1002/asia.201000436.

[14] K. Nidhi, S. Indrajeet, M. Khushboo, K. Gauri, and D. D. J. Sen, "Hydrotropy: A promising tool for solubility enhancement: A review," *International Journal of Drug Development and Research*, vol. 3, pp. 26–33, Apr. 2011.

[15] J. Nanobiotechnol *et al.*, "Self - powered portable melt electrospinning for in situ wound dressing," *Journal of Nanobiotechnology*, pp. 1–10, 2020. DOI: 10.1186/s12951-020-00671-w. [Online]. Available: https://doi.org/10.1186/s12951-020-00671-w.

[16] C. Hacker, Z. Karahaliloglu, G. Seide, E. B. Denkbas, and T. Gries, "Functionally Modified , Melt-Electrospun Thermoplastic Polyurethane Mats for Wound-Dressing Applications," *Journal of Applied Polymer Science*, vol. 40132, pp. 1–12, 2014. DOI: 10.1002/app.40132.

[17] A. Fuchs *et al.*, "Medical-grade polycaprolactone scaffolds made by melt electrospinning writing for oral bone regeneration – a pilot study in vitro," *BMC Oral Health*, pp. 1–11, 2019.

[18] S. Zaiss, T. D. Brown, J. C. Reichert, and A. Berner, "Poly (ϵ -caprolactone) Scaffolds Fabricated by Melt Electrospinning for Bone Tissue Engineering," *Materials*, vol. 9, no. 4, pp. 1–15, 2016. DOI: 10.3390/ma9040232.

[19] G. Hochleitner *et al.*, "Fibre pulsing during melt electrospinning writing," *BioNanoMaterials*, vol. 17, no. 3-4, pp. 159–171, 2016. DOI: 10.1515/bnm-2015-0022.

[20] C. Mingjun *et al.*, "An example of industrialization of melt electrospinning: Polymer melt differential electrospinning," *Advanced Industrial and Engineering Polymer Research*, vol. 2, no. 3, pp. 110–115, 2019. DOI: 10.1016/j.aiepr.2019.06.002. [Online]. Available: https://doi.org/10.1016/j.aiepr.2019.06.002.

[21] G. Hochleitner, F. Chen, C. Blum, P. D. Dalton, B. Amsden, and J. Groll, "Melt electrowriting below the critical translation speed to fabricate crimped elastomer scaffolds with non-linear extension behaviour mimicking that of ligaments and tendons," *Acta Biomaterialia*, vol. 72, pp. 110–120, May 2018. DOI: 10.1016/j.actbio.2018.03.023. [Online]. Available: https://doi.org/10.1016/j.actbio.2018.03.023.

[22] S. Bhat and A. Kumar, "Cell proliferation on three-dimensional chitosan-agarose-gelatin cryogel scaffolds for tissue engineering applications," *Journal of bioscience and bioengineering*, vol. 114, Aug. 2012. DOI: 10.1016/j.jbiosc.2012.07.005.

[23] J. B. Lian and G. S. Stein, "Concepts of osteoblast growth and differentiation: basis for modulation of bone cell development and tissue formation.," *Critical reviews in oral biology and medicine : an official publication of the American Association of Oral Biologists*, vol. 3, no. 3, pp. 269–305, 1992. DOI: 10.1177/10454411920030030501.

[24] S. Bose, S. Vahabzadeh, and A. Bandyopadhyay, "Bone tssue engineering using 3d printing," *Materials Today*, vol. 16, pp. 496–504, Dec. 2013. DOI: 10.1016/j.mattod.2013.11.017.

[25] A. Kumar *et al.*, "Low temperature additive manufacturing of three dimensional scaffolds for bone-tissue engineering applications: Processing related challenges and property assessment," *Materials Science and Engineering: R: Reports*, vol. 103, pp. 1–39, May 2016. DOI: 10.1016/j.mser.2016.01.001.

[26] H. Ding, K. Cao, F. Zhang, W. Boettcher, and R. C. Chang, "A Fundamental Study of Charge Effects on Melt Electrowritten Polymer Fibers," *Materials and Design*, vol. 178, p. 107 857, 2019. DOI: 10.1016/j.matdes.2019.107857. [Online]. Available: https://doi.org/10.1016/j.matdes.2019.107857.

[27] K. Cao, F. Zhang, and R. C. Chang, "A charge-based mechanistic study into the effects of process parameters on fiber accumulating geometry for a melt electrohydrodynamic process," *Processes*, vol. 8, no. 11, p. 1440, Nov. 2020. DOI: 10.3390/pr8111440. [Online]. Available: https://doi.org/10.3390/pr8111440.

[28] M. F. P. Costa and C. Ribeiro, "Parameter estimation of viscoelastic materials: A test case with different optimization strategies," *AIP Conference Proceedings*, vol. 1389, no. September, pp. 771–774, 2011. DOI: 10.1063/1.3636846.

[29] B. Babaei, A. Davarian, K. M. Pryse, E. L. Elson, and G. M. Genin, "Efficient and optimized identification of generalized Maxwell viscoelastic relaxation spectra," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 55, no. November, pp. 32–41, 2016. DOI: 10.1016/j.jmbbm.2015.10.008. [Online]. Available: http://dx.doi.org/10.1016/j.jmbbm.2015.10.008.

[30] M. L. Muerza-Cascante, D. Haylock, D. W. Hutmacher, and P. D. Dalton, "Melt electrospinning and its technologization in tissue engineering," *Tissue Engineering - Part B: Reviews*, vol. 21, no. 2, pp. 187–202, 2015. DOI: 10.1089/ten.teb.2014.0347.

[31] S. Tarafder, W. S. Dernell, A. Bandyopadhyay, and S. Bose, "Sro- and mgo-doped microwave sintered 3d printed tricalcium phosphate scaffolds: Mechanical properties and in vivo osteogenesis in a rabbit model," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 103, no. 3, pp. 679–690, 2015. DOI: https://doi.org/10.1002/jbm.b.33239. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jbm.b.33239. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jbm.b.33239.

[32] S. Pal, "Design of artificial human joints & organs," *Design of Artificial Human Joints & Organs*, vol. 9781461462, pp. 1–419, 2014. DOI: 10.1007/978-1-4614-6255-2.

[33] X. Wang, J. Nyman, X. Dong, H. Leng, and M. Reyes, *Fundamental Biomechanics in Bone Tissue Engineering*. Morgan & Claypool Publishing, 2010, vol. 2, pp. 1–225, ISBN: 9781608451692. DOI: 10.2200/s00246ed1v01y200912tis004.

[34] J. Belinha, "Meshless Methods in Biomechanics: Bone Tissue Remodelling Analysis," *Lecture Notes in Computational Vision and Biomechanics. Volume 8*, vol. 2, p. 328, 2014. DOI: 10.1007/978-94-007-4174-4.

[35] M. Vlasea, A. Basalah, A. Azhari, R. Kandel, and E. Toyserkani, "Additive Manufacturing for Bone Load Bearing Applications," *3D Bioprinting and Nanotechnology in Tissue Engineering and Regenerative Medicine*, pp. 231–263, 2015. DOI: 10.1016/B978-0-12-800547-7.00011-4.

[36] B. AldemIr, S. DIkIcI, O. Karaman, and A. S. Urkmez, "3D Tissue Scaffold Printing On Custom Artificial Bone Applications," *Journal of Natural and Applied Science*, vol. 18, pp. 1–9, 2014.

[37] A. Butscher, M. Bohner, S. Hofmann, L. Gauckler, and R. Müller, "Structural and material approaches to bone tissue engineering in powder-based three-dimensional printing," *Acta biomaterialia*, vol. 7, pp. 907–20, Oct. 2010. DOI: 10.1016/j.actbio.2010.09.039.

[38] H. Pape, A. Evans, and P. Kobbe, "Autologous bone graft: Properties and techniques," *Journal of orthopaedic trauma*, vol. 24 Suppl 1, S36–40, Mar. 2010. DOI: 10.1097/BOT.0b013e3181cec4a1.

[39] K. Zouhary, "Bone graft harvesting from distant sites: Concepts and techniques," *Oral and maxillofacial surgery clinics of North America*, vol. 22, pp. 301–16, v, Aug. 2010. DOI: 10.1016/j.coms.2010.04.007.

[40] B. Leukers *et al.*, "Biocompatibility of ceramic scaffolds for bone replacement made by 3d printing," *Materialwissenschaft und Werkstofftechnik*, vol. 36, pp. 781–787, Dec. 2005. DOI: 10.1002/mawe.200500968.

[41] C. Bergmann *et al.*, "3d printing of bone substitute implants using calcium phosphate and bioactive glasses," *Journal of the European Ceramic Society*, vol. 30, pp. 2563–2567, Sep. 2010. DOI: 10.1016/j.jeurceramsoc.2010.04.037.

[42] M.-O. Montjovent *et al.*, "Biocompatibility of bioresorbable poly(l-lactic acid) composite scaffolds obtained by supercritical gas foaming with human fetal bone cells," *Tissue Engineering*, vol. 11, no. 11-12, pp. 1640–1649, 2005. DOI: 10.1089/ten.2005.11.1640.

[43] G. Brunello *et al.*, "Powder-based 3D printing for bone tissue engineering," *Biotechnology Advances*, vol. 34, no. 5, pp. 740–753, 2016. DOI: 10.1016/j.biotechadv.2016.03.009. [Online]. Available: http://dx.doi.org/10.1016/j.biotechadv.2016.03.009.

[44] H. D. Kim, S. Amirthalingam, S. L. Kim, S. S. Lee, J. Rangasamy, and N. S. Hwang, "Biomimetic materials and fabrication approaches for bone tissue engineering," *Advanced Healthcare Materials*, vol. 6, no. 23, p. 1 700 612, 2017. DOI: https://doi.org/10.1002/adhm.201700612.

[45] A. Leberfinger, D. Ravnic, A. Dhawan, and I. Ozbolat, "Concise review: Bioprinting of stem cells for transplantable tissue fabrication," English (US), *Stem cells translational medicine*, vol. 6, no. 10, pp. 1940–1948, Oct. 2017. DOI: 10.1002/sctm.17-0148.

[46] Y. Huang, X.-F. Zhang, G. Gao, T. Yonezawa, and X. Cui, "3d bioprinting and the current applications in tissue engineering," *Biotechnology Journal*, vol. 12, no. 8, p. 1 600 734, 2017. DOI: https://doi.org/10.1002/biot.201600734. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/biot.201600734. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/biot.201600734.

[47] X. Wang *et al.*, "Topological design and additive manufacturing of porous metals for bone scaffolds and orthopaedic implants: A review," *Biomaterials*, vol. 83, Mar. 2016. DOI: 10.1016/j.biomaterials.2016.01.012.

[48] X.-Y. Zhang, G. Fang, and J. Zhou, "Additively manufactured scaffolds for bone tissue engineering and the prediction of their mechanical behavior: A review," *Materials*, vol. 10, no. 1, 2017. DOI: 10.3390/ma10010050. [Online]. Available: https://www.mdpi.com/1996-1944/10/1/50.

[49] P. S. P. Poh *et al.*, "Polylactides in additive biomanufacturing," *Advanced Drug Delivery Reviews*, vol. 107, pp. 228–246, 2016. DOI: 10.1016/j.addr.2016.07.006. [Online]. Available: http://dx.doi.org/10.1016/j.addr.2016.07.006.

[50] J. Lim, M. You, J. Li, and Z. Li, "Emerging bone tissue engineering via Polyhydroxyalkanoate (PHA)-based scaffolds," *Materials Science and Engineering C*, vol. 79, pp. 917–929, 2017. DOI: 10.1016/j.msec.2017.05.132. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2017.05.132.

[51] S. Wang *et al.*, "Principal theories of electrospinning," *Nanomedicine: Nanotechnology, Biology, and Medicine*, 2017. DOI: 10.1016/j.nano.2016.12.024. [Online]. Available: http://dx.doi.org/10.1016/j.nano.2016.12.024.

[52] F. Obregon, C. Vaquette, S. Ivanovski, D. Hutmacher, and L. Bertassoni, "Three-dimensional bioprinting for regenerative dentistry and craniofacial tissue engineering," *Journal of Dental Research*, vol. 94, no. 9_suppl, 143S–152S, 2015. DOI: 10.1177/0022034515588885.

[53] A. Kantaros, N. Chatzidai, and D. Karalekas, "3d printing-assisted design of scaffold structures," *International Journal of Advanced Manufacturing Technology*, vol. 82, pp. 559–571, Jun. 2016. DOI: 10.1007/s00170-015-7386-6.

[54] P. Warnke *et al.*, "Ceramic scaffolds produced by computer-assisted 3d printing and sintering: Characterization and biocompatibility investigations," *Journal of biomedical materials research. Part B, Applied biomaterials*, vol. 93, pp. 212–7, Apr. 2010. DOI: 10.1002/jbm.b.31577.

[55] M. Bohner and F. Baumgart Robert, "Theoretical model to determine the effects of geometrical factors on the resorption of calcium phosphate bone substitutes," *Biomaterials*, vol. 25, pp. 3569–3582, 2004. DOI: 10.1016/j.biomaterials.2003.10.032.

[56] A. Salerno, M. Oliviero, E. Di Maio, S. Iannace, and P. A. Netti, "Design of porous polymeric scaffolds by gas foaming of heterogeneous blends," *Journal of Materials Science: Materials in Medicine*, vol. 20, no. 10, pp. 2043–2051, 2009. DOI: 10.1007/s10856-009-3767-4.

[57] M. Bohner *et al.*, "Synthesis and characterization of porous b-tricalcium phosphate blocks," *Biomaterials*, vol. 26, pp. 6099–6105, 2005. DOI: 10.1016/j.biomaterials.2005.03.026.

[58] A. Bignon *et al.*, "Effect of micro- and macroporosity of bone substitutes on their mechanical properties and cellular response," *Journal of materials science. Materials in medicine*, vol. 14, pp. 1089–97, Jan. 2004. DOI: 10.1023/B:JMSM.0000004006.90399.b4.

[59] J. X. Lu *et al.*, "Role of interconnections in porous bioceramics on bone recolonization in vitro and in vivo," *JOURNAL OF MATE RIALS SCIENCE: MATERIALS IN ME DI CINE*, vol. 10, pp. 111–120, 1999.

[60] D. W. Hutmacher, "Sca!olds in tissue engineering bone and cartilage," *Biomaterials*, vol. 21, pp. 2529–2543, 2000.

[61] I. Armentano, M. Dottori, E. Fortunati, S. Mattioli, and J. Kenny, "Biodegradable polymer matrix nanocomposites for tissue engineering: A review," *Polymer Degradation and Stability*, vol. 95, pp. 2126–2146, Nov. 2010. DOI: 10.1016/j.polymdegradstab.2010.06.007.

[62] L. Nair and C. Laurencin, "Biodegradable polymers as biomaterials," *Progress in Polymer Science*, vol. 32, pp. 762–798, Aug. 2007. DOI: 10.1016/j.progpolymsci.2007.05.017.

[63] M. Sawkins, P. Mistry, B. Brown, K. Shakesheff, L. Bonassar, and J. Yang, "Cell and protein compatible 3d bioprinting of mechanically strong constructs for bone repair," *Biofabrication*, vol. 7, Sep. 2015. DOI: 10.1088/1758-5090/7/3/035004.

[64] A. Zima, "Hydroxyapatite-chitosan based bioactive hybrid biomaterials with improved mechanical strength," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 193, pp. 175–184, 2018. DOI: 10.1016/j.saa.2017.12.008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1386142517309812.

[65] H. Zhou and J. Lee, "Nanoscale hydroxyapatite particles for bone tissue engineering," *Acta biomaterialia*, vol. 7, pp. 2769–81, Mar. 2011. DOI: 10.1016/j.actbio.2011.03.019.

[66] A. Oliveira, I. Leonor, P. Malafaya, C. Alves, H. Azevedo, and R. Reis, "Tailoring the bioactivity of natural origin inorganic - Polymeric based systems," *Key Engineering Materials*, vol. 240-242, pp. 111–142, 2003. DOI: 10.4028/www.scientific.net/KEM.240-242.111.

[67] E. Skwarek, O. Goncharuk, D. Sternik, W. Janusz, K. Gdula, and V. M. Gun'ko, "Synthesis, Structural, and Adsorption Properties and Thermal Stability of Nanohydroxyapatite/Polysaccharide Composites," *Nanoscale Research Letters*, vol. 12, no. 1, p. 155, 2017. DOI: 10.1186/s11671-017-1911-5. [Online]. Available: http://nanoscalereslett.springeropen.com/articles/10.1186/s11671-017-1911-5.

[68] M. Shakir, R. Jolly, M. S. Khan, A. Rauf, and S. Kazmi, "Nano-hydroxyapatite/??-CD/chitosan nanocomposite for potential applications in bone tissue engineering," *International Journal of Biological Macromolecules*, vol. 93, pp. 276–289, 2016. DOI: 10.1016/j.ijbiomac.2016.08.046. [Online]. Available: http://dx.doi.org/10.1016/j.ijbiomac.2016.08.046.

[69] M. Shakir, R. Jolly, M. S. Khan, N. E. Iram, T. K. Sharma, and S. I. Al-Resayes, "Synthesis and characterization of a nano-hydroxyapatite/chitosan/polyethylene glycol nanocomposite for bone tissue engineering," *Polymers for Advanced Technologies*, vol. 26, no. 1, pp. 41–48, 2015. DOI: 10.1002/pat.3417.

[70] S. Petricca, K. Marra, and P. Kumta, "Chemical synthesis of poly(lactic-co-glycolic acid)/hydroxyapatite composites for orthopaedic applicatons," *Acta biomaterialia*, vol. 2, pp. 277–86, Jun. 2006. DOI: 10.1016/j.actbio.2005.12.004.

[71] C. Sarkar, P. Kumari, K. Anuvrat, S. K. Sahu, J. Chakraborty, and S. Garai, "Synthesis and characterization of mechanically strong carboxymethyl cellulose–gelatin–hydroxyapatite nanocomposite for load-bearing orthopedic application," *Journal of Materials Science*, vol. 53, no. 1, pp. 230–246, 2018. DOI: 10.1007/s10853-017-1528-1.

[72] G. Zhou *et al.*, "Innovative biodegradable poly(L-lactide)/collagen/hydroxyapatite composite fibrous scaffolds promote osteoblastic proliferation and differentiation," *International Journal of Nanomedicine*, vol. 12, pp. 7577–7588, 2017. DOI: 10.2147/IJN.S146679.

[73] K. Rezwan, Q. Chen, J. Blaker, and A. Boccaccini, "Biodegradable and bioactive porous polymer/inorganic composite scaffolds for bone tissue engineering," *Biomaterials*, vol. 27, pp. 3413–31, Jul. 2006. DOI: 10.1016/j.biomaterials.2006.01.039.

[74] D. Tang, R. Tare, L.-Y. Yang, D. Williams, K.-L. Ou, and R. Oreffo, "Biofabrication of bone tissue: Approaches, challenges and translation for bone regeneration," *Biomaterials*, vol. 83, Jan. 2016. DOI: 10.1016/j.biomaterials.2016.01.024.

[75] G. Webler *et al.*, "Use of micrometric latex beads to improve the porosity of hydroxyapatite obtained by chemical coprecipitation method," *Applied Surface Science*, vol. 436, no. 3, pp. 141–151, 2017. DOI: 10.1016/j.apsusc.2017.11.218. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169433217335110.

[76] A. Aravamudhan, D. M. Ramos, A. A. Nada, and S. G. Kumbar, *Natural Polymers: Polysaccharides and Their Derivatives for Biomedical Applications*. Elsevier Inc., 2014, pp. 67–89, ISBN: 9780123969835. DOI: 10.1016/B978-0-12-396983-5.00004-1. arXiv: arXiv:1011.1669v3. [Online]. Available: http://dx.doi.org/10.1016/B978-0-12-396983-5.00004-1.

[77] G. Cama, *1 - Calcium phosphate cements for bone regeneration*. Woodhead Publishing Limited, 2014, pp. 3–25, ISBN: 9780857098108. DOI: 10.1533/9780857098104.1.3. [Online]. Available: http://dx.doi.org/10.1533/9780857098104.1.3.

[78] K. Chawla, *Biomaterials for Tissue Engineering*. Humana Press, 2018, ISBN: 9781493977390.

[79] D.-W. Cho, J.-S. Lee, J. Jang, J. W. Jung, J. H. Park, and F. Pati, *Tissue engineering: osteochondral tissue* (2053-2571). Morgan & Claypool Publishers, 2015, 11–1 to 11–6, ISBN: 978-1-6817-4079-9. DOI: 10.1088/978-1-6817-4079-9ch11. [Online]. Available: https://dx.doi.org/10.1088/978-1-6817-4079-9ch11.

[80] M. Niaounakis, *Chapter 1 - Introduction*. Oxford: William Andrew Publishing, 2015, pp. 1–77, ISBN: 978-0-323-26698-7. DOI: https://doi.org/10.1016/B978-0-323-26698-7.00001-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323266987000015.

[81] F. R. Rolando Chamy, *Biodegradation - Life of Science*. Rijeka: Intech Open., 2013. DOI: 10.5772/52777. [Online]. Available: https://doi.org/10.5772/52777.

[82] S. Stratton, N. B. Shelke, K. Hoshino, S. Rudraiah, and S. G. Kumbar, "Bioactive polymeric scaffolds for tissue engineering," *Bioactive Materials*, vol. 1, no. 2, pp. 93–108, Dec. 2016. DOI: 10.1016/j.bioactmat.2016.11.001. [Online]. Available: https://doi.org/10.1016/j.bioactmat.2016.11.001.

[83] B. Lukasiewicz, P. Basnett, R. Nigmatullin, R. Matharu, J. C. Knowles, and I. Roy, "Binary polyhydroxyalkanoate systems for soft tissue engineering," *Acta Biomaterialia*, vol. 71, pp. 225–234, 2018. DOI: 10.1016/j.actbio.2018.02.027. [Online]. Available: https://doi.org/10.1016/j.actbio.2018.02.027.

[84] Z. A. Raza, S. Abid, and I. M. Banat, "Polyhydroxyalkanoates: Characteristics, production, recent developments and applications," *International Biodeterioration and Biodegradation*, vol. 126, no. September 2017, pp. 45–56, 2018. DOI: 10.1016/j.ibiod.2017.10.001.

[85] S. Saska *et al.*, "Three-dimensional printing and in vitro evaluation of poly(3-hydroxybutyrate) scaffolds functionalized with osteogenic growth peptide for tissue engineering," *Materials Science and Engineering: C*, vol. 89, no. March 2017, pp. 265–273, 2018. DOI: https://doi.org/10.1016/j.msec.2018.04.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0928493117309943.

[86] S. Misra, S. Valappil, I. Roy, and A. Boccaccini, "Polyhydroxyalkanoate (pha)/inorganic phase composites for tissue engineering applications," *Biomacromolecules*, vol. 7, pp. 2249–58, Sep. 2006. DOI: 10.1021/bm060317c.

[87] H. Chang *et al.*, "Poly(3-hydroxybutyrate-co-3-hydroxyhexanoate)-based scaffolds for tissue engineering," *Brazilian Journal of Medical and Biological Research*, vol. 47, no. 7, pp. 533–539, 2014. DOI: 10.1590/1414-431X20143930. [Online]. Available: http://www.scielo.br/scielo.php?script=sci%7B%5C_%7Darttext%7B%5C&%7Dpid=S0100-879X2014000700533%7B%5C&%7Dlng=en%7B%5C&%7Dnrm=iso%7B%5C&%7Dtlng=en.

[88] Y.-J. Hu, X. Wei, W. Zhao, Y.-S. Liu, and G.-Q. Chen, "Biocompatibility of poly(3-hydroxybutyrate-co-3-hydroxyvalerate-co-3-hydroxyhexanoate) with bone marrow mesenchymal stem cells," *Acta Biomaterialia*, vol. 5, no. 4, pp. 1115–1125, 2009. DOI: https://doi.org/10.1016/j.actbio.2008.09.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742706108003012.

[89] M. Okamoto and B. John, "Synthetic biopolymer nanocomposites for tissue engineering scaffolds," *Progress in Polymer Science*, vol. 38, pp. 1487–1503, Oct. 2013. DOI: 10.1016/j.progpolymsci.2013.06.001.

[90] A. Butscher, M. Bohner, N. Döbelin, S. Hofmann, and R. Müller, "New depowdering-friendly designs for three-dimensional printing of calcium phosphate bone substitutes," *Acta biomaterialia*, vol. 9, Jul. 2013. DOI: 10.1016/j.actbio.2013.07.019.

[91] C. Ruiz-Aguilar, E. A. Aguilar-Reyes, M. Flores-Martínez, C. A. León-Patiño, and R. E. Nuñéz-Anita, "Synthesis and characterisation of β-TCP/bioglass/zirconia scaffolds," *Advances in Applied Ceramics*, vol. 116, no. 8, pp. 452–461, 2017. DOI: 10.1080/17436753.2017.1356043.

[92] R. Hambli, "Micro-ct finite element model and experimental validation of trabecular bone damage and fracture," *Bone*, vol. 56, Jul. 2013. DOI: 10.1016/j.bone.2013.06.028.

[93] S. J. Schambach, S. Bag, L. Schilling, C. Groden, and M. A. Brockmann, "Application of micro-CT in small animal imaging," *Methods*, vol. 50, no. 1, pp. 2–13, Jan. 2010. DOI: 10.1016/j.ymeth.2009.08.007. [Online]. Available: https://doi.org/10.1016/j.ymeth.2009.08.007.

[94] D. Sreenivasan *et al.*, "Computer modelling integrated with micro-CT and material testing provides additional insight to evaluate bone treatments: Application to a beta-glycan derived whey protein mice model," *Computers in Biology and Medicine*, vol. 68, pp. 9–20, 2016. DOI: 10.1016/j.compbiomed.2015.10.017.

[95] L. Qin, *Advanced bioimaging technologies in assessment of the quality of bone and scaffold materials. [electronic resource] : techniques and applications.* Springer, 2007. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=cat03710a%5C&AN=alb.7950245%5C&site=eds-live%5C&scope=site.

[96] H. S. Tuan and D. W. Hutmacher, "Application of micro CT and computation modeling in bone tissue engineering," *Computer-Aided Design*, vol. 37, no. 11, pp. 1151–1161, Sep. 2005. DOI: 10.1016/j.cad.2005.02.006. [Online]. Available: https://doi.org/10.1016/j.cad.2005.02.006.

[97] M. Juneja, N. Thakur, D. Kumar, A. Gupta, B. Bajwa, and P. Jindal, "Accuracy in dental surgical guide fabrication using different 3-D printing techniques," *Additive Manufacturing*, vol. 22, no. September 2017, pp. 243–255, 2018. DOI: 10.1016/j.addma.2018.05.012. [Online]. Available: https://doi.org/10.1016/j.addma.2018.05.012.

[98] S. Rianmora and P. Koomsap, "Recommended slicing positions for adaptive direct slicing by image processing technique," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 9-12, pp. 1021–1033, Jul. 2009. DOI: 10.1007/s00170-009-2162-0. [Online]. Available: https://doi.org/10.1007/s00170-009-2162-0.

[99] J. Gardan, "Additive manufacturing technologies: State of the art and trends," *International Journal of Production Research*, vol. 54, no. 10, pp. 3118–3132, Nov. 2015. DOI: 10.1080/00207543.2015.1115909. [Online]. Available: https://doi.org/10.1080/00207543.2015.1115909.

[100] W. Oropallo and L. A. Piegl, "Ten challenges in 3d printing," *Engineering with Computers*, vol. 32, no. 1, pp. 135–148, Jun. 2015. DOI: 10.1007/s00366-015-0407-0. [Online]. Available: https://doi.org/10.1007/s00366-015-0407-0.

[101] M. R. Dias, J. M. Guedes, C. L. Flanagan, S. J. Hollister, and P. R. Fernandes, "Optimization of scaffold design for bone tissue engineering: A computational and experimental study," *Medical Engineering and Physics*, vol. 36, no. 4, pp. 448–457, Apr. 2014. DOI: 10.1016/j.medengphy.2014.02.010. [Online]. Available: https://doi.org/10.1016/j.medengphy.2014.02.010.

[102] S. Hollister, R. Maddox, and J. Taboas, "Optimal design and fabrication of scaffolds to mimic tissue properties and satisfy biological constraints," *Biomaterials*, vol. 23, no. 20, pp. 4095–4103, Oct. 2002. DOI: 10.1016/s0142-9612(02)00148-5. [Online]. Available: https://doi.org/10.1016/s0142-9612(02)00148-5.

[103] S. J. Hollister and C. Y. Lin, "Computational design of tissue engineering scaffolds," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 31-32, pp. 2991–2998, Jun. 2007. DOI: 10.1016/j.cma.2006.09.023. [Online]. Available: https://doi.org/10.1016/j.cma.2006.09.023.

[104] D. J. Yoo, "Advanced projection image generation algorithm for fabrication of a tissue scaffold using volumetric distance field," *International Journal of Precision Engineering and Manufacturing*, vol. 15, no. 10, pp. 2117–2126, 2014. DOI: 10.1007/s12541-014-0571-y.

[105] W.-H. Chen, Y.-Y. Liu, F.-H. Zhang, Y.-Z. Yu, H.-P. Chen, and Q.-X. Hu, "Osteochondral integrated scaffolds with gradient structure by 3d printing forming," *International Journal of Automation and Computing*, vol. 12, no. 2, pp. 220–228, Apr. 2015. DOI: 10.1007/s11633-014-0853-y. [Online]. Available: https://doi.org/10.1007/s11633-014-0853-y.

[106] N. Yang, C.-f. Du, S. Wang, Y. Yang, and C. Zhang, "Mathematically defined gradient porous materials," *Materials Letters*, vol. 173, pp. 136–140, Jun. 2016. DOI: 10.1016/j.matlet.2016.03.021. [Online]. Available: https://doi.org/10.1016/j.matlet.2016.03.021.

[107] N. Yang, Y. Tian, and D. Zhang, "Novel real function based method to construct heterogeneous porous scaffolds and additive manufacturing for use in medical engineering," *Medical Engineering and Physics*, vol. 37, no. 11, pp. 1037–1046, Nov. 2015. DOI: 10.1016/j.medengphy.2015.08.006. [Online]. Available: https://doi.org/10.1016/j.medengphy.2015.08.006.

[108] N. Yang and K. Zhou, "Effective method for multi-scale gradient porous scaffold design and fabrication," *Materials Science and Engineering C*, vol. 43, pp. 502–505, 2014. DOI: 10.1016/j.msec.2014.07.052. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2014.07.052.

[109] A. L. Hong *et al.*, "Feasibility of fabricating personalized 3D-printed bone grafts guided by high-resolution imaging," *Imaging Informatics for Healthcare*, vol. 10138, T. S. Cook and J. Zhang, Eds., pp. 153–158, 2017. DOI: 10.1117/12.2254475. [Online]. Available: https://doi.org/10.1117/12.2254475.

[110] J. J. Yoon and T. G. Park, "Degradation behaviors of biodegradable macroporous scaffolds prepared by gas foaming of effervescent salts," *Journal of Biomedical Materials Research*, vol. 55, no. 3, pp. 401–408, 2001. DOI: 10.1002/1097-4636(20010605)55:3<401::AID-JBM1029>3.0.CO;2-H.

[111] A. R. Boccaccini and V. Maquet, "Bioresorbable and bioactive polymer/Bioglass?? composites with tailored pore structure for tissue engineering applications," *Composites Science and Technology*, vol. 63, no. 16, pp. 2417–2429, 2003. DOI: 10.1016/S0266-3538(03)00275-6.

[112] M. Z. Moghadam, S. Hassanajili, F. Esmaeilzadeh, M. Ayatollahi, and M. Ahmadi, "Formation of porous HPCL/LPCL/HA scaffolds with supercritical CO2gas foaming method," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 69, no. August 2016, pp. 115–127, 2017. DOI: 10.1016/j.jmbbm.2016.12.014. [Online]. Available: http://dx.doi.org/10.1016/j.jmbbm.2016.12.014.

[113] A. Salerno, J. Saurina, and C. Domingo, "Supercritical CO2 foamed polycaprolactone scaffolds for controlled delivery of 5-fluorouracil, nicotinamide and triflusal," *International journal of pharmaceutics*, vol. 496, no. 2, pp. 654–663, 2015. DOI: 10.1016/j.ijpharm.2015.11.012.

[114] C. Zhou *et al.*, "Combination of fused deposition modeling and gas foaming technique to fabricated hierarchical macro/microporous polymer scaffolds," *Materials and Design*, vol. 109, pp. 415–424, 2016. DOI: 10.1016/j.matdes.2016.07.094. [Online]. Available: http://dx.doi.org/10.1016/j.matdes.2016.07.094.

[115] a. Salerno, P. A. Netti, E. Di Maio, and S. Iannace, "Engineering of Foamed Structures for Biomedical Application," *Journal of Cellular Plastics*, vol. 45, no. 2, pp. 103–117, 2009. DOI: 10.1177/0021955X08099929.

[116] M. Costantini *et al.*, "Correlation between porous texture and cell seeding efficiency of gas foaming and microfluidic foaming scaffolds," *Materials Science and Engineering C*, vol. 62, pp. 668–677, 2016. DOI: 10.1016/j.msec.2016.02.010.

[117] Y. S. Nam, J. J. Yoon, and T. G. Park, "A novel fabrication method of macroporous biodegradable polymer scaffolds using gas foaming salt as a porogen additive," *Journal of Biomedical Materials Research*, vol. 53, no. 1, pp. 1–7, 2000. DOI: 10.1002/(SICI)1097-4636(2000)53:1<1::AID-JBM1>3.0.CO;2-R.

[118] X. Jing, H. Y. Mi, and L. S. Turng, "Comparison between PCL/hydroxyapatite (HA) and PCL/halloysite nanotube (HNT) composite scaffolds prepared by co-extrusion and gas foaming," *Materials Science and Engineering C*, vol. 72, pp. 53–61, 2017. DOI: 10.1016/j.msec.2016.11.049. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2016.11.049.

[119] T. Y. Bak, M. S. Kook, S. C. Jung, and B. H. Kim, "Biological effect of gas plasma treatment on CO2 gas foaming/salt leaching fabricated porous polycaprolactone scaffolds in bone tissue engineering," *Journal of Nanomaterials*, vol. 2014, 2014. DOI: 10.1155/2014/657542.

[120] B. M. Carvalho *et al.*, "Cryogel poly(acrylamide): Synthesis, structure and applications," *Separation and Purification Reviews*, vol. 43, no. 3, pp. 241–262, 2014. DOI: 10.1080/15422119.2013.795902.

[121] A. Kumar, R. Mishra, Y. Reinwald, and S. Bhat, "Cryogels: Freezing unveiled by thawing," *Materials Today*, vol. 13, no. 11, pp. 42–44, 2010. DOI: 10.1016/S1369-7021(10)70202-9.

[122] O. Okay, *Polymeric cryogels macroporous gels with remarkable properties*. Springer International Publishing, 2014, vol. 263, pp. iii–iv, ISBN: 9783319058450. DOI: 10.1007/978-3-319-05846-7.

[123] S. C. Rodrigues, C. L. Salgado, A. Sahu, M. P. Garcia, M. H. Fernandes, and F. J. Monteiro, "Preparation and characterization of collagen-nanohydroxyapatite biocomposite scaffolds by cryogelation method for bone tissue engineering applications," *Journal of Biomedical Materials Research - Part A*, vol. 101 A, no. 4, pp. 1080–1094, 2013. DOI: 10.1002/jbm.a.34394.

[124] N. Bölgen *et al.*, "Stem cell suspension injected HEMA-lactate-dextran cryogels for regeneration of critical sized bone defects," *Artificial Cells, Nanomedicine and Biotechnology*, vol. 42, no. 1, pp. 70–77, 2014. DOI: 10.3109/21691401.2013.775578.

[125] M. E. Han *et al.*, "Extracellular matrix-based cryogels for cartilage tissue engineering," *International Journal of Biological Macromolecules*, vol. 93, pp. 1410–1419, 2016. DOI: 10.1016/j.ijbiomac.2016.05.024. [Online]. Available: http://dx.doi.org/10.1016/j.ijbiomac.2016.05.024.

[126] R. Mishra and A. Kumar, "Inorganic/organic biocomposite cryogels for regeneration of bony tissues," *Journal of Biomaterials Science, Polymer Edition*, vol. 22, no. 16, pp. 2107–2126, 2011. DOI: 10.1163/092050610X534230.

[127] J. Van Rie *et al.*, "Cryogel-PCL combination scaffolds for bone tissue repair," *Journal of Materials Science: Materials in Medicine*, vol. 26, no. 3, 2015. DOI: 10.1007/s10856-015-5465-8.

[128] D. Singh, A. Tripathi, V. Nayak, and A. Kumar, "Proliferation of chondrocytes on a 3-d modelled macroporous poly(hydroxyethyl methacrylate)-gelatin cryogel," *Journal of Biomaterials Science, Polymer Edition*, vol. 22, no. 13, pp. 1733–1751, 2011. DOI: 10.1163/092050610X522486.

[129] C. L. Salgado, L. Grenho, M. H. Fernandes, B. J. Colaço, and F. J. Monteiro, "Biodegradation, biocompatibility, and osteoconduction evaluation of collagen-nanohydroxyapatite cryogels for bone tissue regeneration," *Journal of Biomedical Materials Research - Part A*, vol. 104, no. 1, pp. 57–70, 2016. DOI: 10.1002/jbm.a.35540.

[130] D. B. Raina, H. Isaksson, A. K. Teotia, L. Lidgren, M. Tägil, and A. Kumar, "Biocomposite macroporous cryogels as potential carrier scaffolds for bone active agents augmenting bone regeneration," *Journal of Controlled Release*, vol. 235, pp. 365–378, 2016. DOI: 10.1016/j.jconrel.2016.05.061. [Online]. Available: http://dx.doi.org/10.1016/j.jconrel.2016.05.061.

[131] B. Yetiskin, C. Akinci, and O. Okay, "Cryogelation within cryogels: Silk fibroin scaffolds with single-, double- and triple-network structures," *Polymer (United Kingdom)*, vol. 128, pp. 47–56, 2017. DOI: 10.1016/j.polymer.2017.09.023. [Online]. Available: https://doi.org/10.1016/j.polymer.2017.09.023.

[132] J. Hendriks, J. Riesle, and C. A. van Blitterswijk, "Co-culture in cartilage tissue engineering," *Journal of tissue engineering and regenerative medicine*, vol. 4, no. 7, pp. 524–531, 2010. DOI: 10.1002/term. arXiv: NIHMS150003.

[133] D. Kai, M. P. Prabhakaran, B. Stahl, M. Eblenkamp, E. Wintermantel, and S. Ramakrishna, "Mechanical properties and in vitro behavior of nanofiberhydrogel composites for tissue engineering applications," *Nanotechnology*, vol. 23, no. 9, 2012. DOI: 10.1088/0957-4484/23/9/095705.

[134] B. Y. Ozturk *et al.*, "The treatment of segmental bone defects in rabbit tibiae with vascular endothelial growth factor (VEGF)-loaded gelatin/hydroxyapatite "cryogel" scaffold," *European Journal of Orthopaedic Surgery and Traumatology*, vol. 23, no. 7, pp. 767–774, 2013. DOI: 10.1007/s00590-012-1070-4.

[135] L. Fassina *et al.*, "Ultrasound stimulus to enhance the bone regeneration capability of gelatin cryogels," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013, pp. 846–849. DOI: 10.1109/EMBC.2013.6609633.

[136] L. Fassina *et al.*, "Electromagnetic stimulation to optimize the bone regeneration capacity of gelatin-based cryogels," *International Journal of Immunopathology and Pharmacology*, vol. 25, no. 1, pp. 165–174, 2012. DOI: 10.1177/039463201202500119.

[137] A. Golunova, J. Jaroš, V. Jurtíková, I. Kotelnikov, and J. Kotek, "N - (2-Hydroxypropyl) Methacrylamide Based Cryogels – Synthesis and Biomimetic Modification for Stem Cell Applications," *Physiological Research*, vol. 64, 2015.

[138] K. R. Hixon *et al.*, "The calcification potential of cryogel scaffolds incorporated with various forms of hydroxyapatite for bone regeneration," *Biomedical Materials*, vol. 12, no. 2, p. 025 005, Mar. 2017. DOI: 10.1088/1748-605x/aa5d76. [Online]. Available: https://doi.org/10.1088/1748-605x/aa5d76.

[139] K. (1.) Hixon, C. (1.) Eberlin, P. (1.) Kadakia, E. (1.) Jain, S. (1.) Sell, and S. (2.) McBride-Gagyi, "A comparison of cryogel scaffolds to identify an appropriate structure for promoting bone regeneration.," *Biomedical Physics and Engineering Express*, vol. 2, no. 3, 2016. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85011354170%5C&site=eds-live%5C&scope=site.

[140] N. Bolgen, Y. Yang, P. Korkusuz, E. Guzel, A. J. El Haj, and E. Piskin, "Three-dimensional ingrowth of bone cells within biodegradable cryogel scaffolds in bioreactors at different regimes.," *TISSUE ENGINEERING PART A*, vol. 14, no. 10, pp. 1743–1750, 2008. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edswsc%5C&AN=000259998000015%5C&site=eds-live%5C&scope=site.

[141] H. (1.) Kim *et al.*, "Enhanced osteogenic commitment of human mesenchymal stem cells on polyethylene glycol-based cryogel with graphene oxide substrate.," *ACS Biomaterials Science and Engineering*, vol. 3, no. 10, pp. 2470–2479, 2017. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85030830446%5C&site=eds-live%5C&scope=site.

[142] N. Kemence and N. Bolgen, "Gelatin- and hydroxyapatite-based cryogels for bone tissue engineering: Synthesis, characterization, in vitro and in vivo biocompatibility.," *JOURNAL OF TISSUE ENGINEERING AND REGENERATIVE MEDICINE*, vol. 11, no. 1, pp. 20–33, 2017. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edswsc%5C&AN=000394173600002%5C&site=eds-live%5C&scope=site.

[143] C. Liu *et al.*, "RSC Advances application in bone tissue engineering †," *RSC Advances*, vol. 5, pp. 20 227–20 233, 2015. DOI: 10.1039/C4RA15893H. [Online]. Available: http://dx.doi.org/10.1039/C4RA15893H.

[144] 3.) Mishra R. (1, 2.) Raina D.B. (1, A. (1.) Kumar, M. (2.) Pelkonen, L. (2.) Lidgren, and M. (2.) Tägil, "Study of in vitro and in vivo bone formation in composite cryogels and the influence of electrical stimulation.," *International Journal of Biological Sciences*, vol. 11, no. 11, pp. 1325–1336, 2015. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-84945309162%5C&site=eds-live%5C&scope=site.

[145] M. (1.) Vetrik *et al.*, "Porous heat-treated polyacrylonitrile scaffolds for bone tissue engineering.," *ACS Applied Materials and Interfaces*, vol. 10, no. 10, pp. 8496–8506, 2018. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85043780531%5C&site=eds-live%5C&scope=site.

[146] E. Saino *et al.*, "Effects of electromagnetic stimulation on osteogenic differentiation of human mesenchymal stromal cells seeded onto gelatin cryogel," *International Journal of Immunopathology and Pharmacology*, vol. 24, no. 1, pp. 1–6, 2011. DOI: 10.1177/03946320110241S201.

[147] ISO, "Additive manufacturing – General principles – Part 2: Overview of process categories and feedstock," International Organization for Standardization, Geneva, CH, Standard, Jan. 2015.

[148] H. E. Jazayeri *et al.*, "The cross-disciplinary emergence of 3D printed bioceramic scaffolds in orthopedic bioengineering," *Ceramics International*, vol. 44, no. 1, pp. 1–9, 2018. DOI: 10.1016/j.ceramint.2017.09.095. [Online]. Available: https://doi.org/10.1016/j.ceramint.2017.09.095.

[149] ISO, "Additive manufacturing – General principles – Fundamentals and vocabulary," International Organization for Standardization, Geneva, CH, Standard, Nov. 2021.

[150] "Additive manufacturing - general principles - fundamentals and vocabulary (iso/astm 52900:2021)," International Organization for Standardization, Geneva, CH, Standard, Mar. 2020.

[151] A.-V. Do, B. Khorsand, S. M. Geary, and A. K. Salem, "3D Printing of Scaffolds for Tissue Regeneration Applications," *Advanced Healthcare Materials*, vol. 4, no. 12, pp. 1742–1762, 2015. DOI: 10.1002/adhm.201500168. [Online]. Available: http://doi.wiley.com/10.1002/adhm.201500168.

[152] J. L. Dávila, M. S. Freitas, P. I. Neto, Z. C. Silveira, J. V. L. Silva, and M. A. D'Ávila, "Software to generate 3-D continuous printing paths for the fabrication of tissue engineering scaffolds," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 5-8, pp. 1671–1677, 2016. DOI: 10.1007/s00170-015-7866-8.

[153] C. Han *et al.*, "Continuous functionally graded porous titanium scaffolds manufactured by selective laser melting for bone implants," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 80, no. January, pp. 119–127, 2018. DOI: 10.1016/j.jmbbm.2018.01.013. [Online]. Available: https://doi.org/10.1016/j.jmbbm.2018.01.013.

[154] A. Bakhshian Nik and B. Vahidi, "The effect of bone scaffold gradient architecture design on stem cell mechanical modulation: A computational study," in *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*, 2015, pp. 309–313. DOI: 10.1109/ICBME.2015.7404161.

[155] L. Xiao and W. Song, "Additively-manufactured functionally graded Ti-6Al-4V lattice structures with high strength under static and dynamic loading: Experiments," *International Journal of Impact Engineering*, vol. 111, no. September 2017, pp. 255–272, 2018. DOI: 10.1016/j.ijimpeng.2017.09.018.

[156] J. A. Inzana *et al.*, "3D printing of composite calcium phosphate and collagen scaffolds for bone regeneration," *Biomaterials*, vol. 35, no. 13, pp. 4026–4034, 2014. DOI: 10.1016/j.biomaterials.2014.01.064. arXiv: NIHMS150003. [Online]. Available: http://dx.doi.org/10.1016/j.biomaterials.2014.01.064.

[157] B. Utela, D. Storti, R. Anderson, and M. Ganter, "A review of process development steps for new material systems in three dimensional printing (3DP)," *Journal of Manufacturing Processes*, vol. 10, no. 2, pp. 96–104, 2008. DOI: 10.1016/j.jmapro.2009.03.002. [Online]. Available: http://dx.doi.org/10.1016/j.jmapro.2009.03.002.

[158] J. Will *et al.*, "Porous ceramic bone scaffolds for vascularized bone tissue regeneration," *Journal of Materials Science: Materials in Medicine*, vol. 19, no. 8, pp. 2781–2790, 2008. DOI: 10.1007/s10856-007-3346-5.

[159] Z. Zhou, F. Buchanan, C. Mitchell, and N. Dunne, "Printability of calcium phosphate: Calcium sulfate powders for the application of tissue engineered bone scaffolds using the 3D printing technique," *Materials Science and Engineering C*, vol. 38, no. 1, pp. 1–10, 2014. DOI: 10.1016/j.msec.2014.01.027. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2014.01.027.

[160] H. Chen, Q. Wei, S. Wen, Z. Li, and Y. Shi, "Flow behavior of powder particles in layering process of selective laser melting: Numerical modeling and experimental verification based on discrete element method," *International Journal of Machine Tools and Manufacture*, vol. 123, no. September, pp. 146–159, 2017. DOI: 10.1016/j.ijmachtools.2017.08.004. [Online]. Available: https://doi.org/10.1016/j.ijmachtools.2017.08.004.

[161] A. Torkkeli, "Droplet microfluidics on a planar surface," *VTT Publications*, vol. 63, no. 504, pp. 3–194, 2003. DOI: 10.1002/aic. arXiv: 0201037v1 [arXiv:physics].

[162] R. Vasireddi and B. Basu, "Conceptual design of three-dimensional scaffolds of powder-based materials for bone tissue engineering applications," *Rapid Prototyping Journal*, vol. 21, no. 6, pp. 716–724, 2015. DOI: 10.1108/RPJ-12-2013-0123. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-12-2013-0123.

[163] J. P. Temple *et al.*, "Engineering anatomically shaped vascularized bone grafts with hASCs and 3D-printed PCL scaffolds," *Journal of Biomedical Materials Research - Part A*, vol. 102, no. 12, pp. 4317–4325, 2014. DOI: 10.1002/jbm.a.35107.

[164] Q. Wei, Y. Wang, W. Chai, Y. Zhang, and X. Chen, "Molecular dynamics simulation and experimental study of the bonding properties of polymer binders in 3D powder printed hydroxyapatite bioceramic bone scaffolds," *Ceramics International*, vol. 43, no. 16, pp. 13 702–13 709, 2017. DOI: 10.1016/j.ceramint.2017.07.082. [Online]. Available: http://dx.doi.org/10.1016/j.ceramint.2017.07.082.

[165] Q. Wei *et al.*, "Study the bonding mechanism of binders on hydroxyapatite surface and mechanical properties for 3DP fabrication bone scaffolds," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 57, pp. 190–200, 2016. DOI: 10.1016/j.jmbbm.2015.12.007. [Online]. Available: http://dx.doi.org/10.1016/j.jmbbm.2015.12.007.

[166] G. Cadafalch Gazquez, H. Chen, L. Moroni, B. A. Boukamp, and J. E. ten Elshof, "β-Tricalcium phosphate nanofiber scaffolds with fine unidirectional grains," *Materials Letters*, vol. 208, pp. 118–121, 2017. DOI: 10.1016/j.matlet.2017.05.038. [Online]. Available: http://dx.doi.org/10.1016/j.matlet.2017.05.038.

[167] Y. Deng *et al.*, "3D printed scaffolds of calcium silicate-doped β-TCP synergize with co-cultured endothelial and stromal cells to promote vascularization and bone formation," *Scientific Reports*, vol. 7, no. 1, pp. 1–14, 2017. DOI: 10.1038/s41598-017-05196-1. [Online]. Available: http://dx.doi.org/10.1038/s41598-017-05196-1.

[168] M. Castilho *et al.*, "Fabrication of computationally designed scaffolds by low temperature 3d printing," *Biofabrication*, vol. 5, no. 3, p. 035 012, Jul. 2013. DOI: 10.1088/1758-5082/5/3/035012. [Online]. Available: https://doi.org/10.1088/1758-5082/5/3/035012.

[169] M. Seidenstuecker *et al.*, "3D Powder Printed Bioglass and β-Tricalcium Phosphate Bone Scaffolds," *Materials*, vol. 11, no. 1, p. 13, 2017. DOI: 10.3390/ma11010013. [Online]. Available: http://www.mdpi.com/1996-1944/11/1/13.

[170] H. Shao *et al.*, "Custom Repair of Mandibular Bone Defects with 3D Printed Bioceramic Scaffolds," *Journal of Dental Research*, p. 002 203 451 773 484, 2017. DOI: 10.1177/0022034517734846. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0022034517734846.

[171] R. Trombetta, J. A. Inzana, E. M. Schwarz, S. L. Kates, and H. A. Awad, "3D Printing of Calcium Phosphate Ceramics for Bone Tissue Engineering and Drug Delivery," *Annals of Biomedical Engineering*, vol. 45, no. 1, pp. 1–22, 2016. DOI: 10.1007/s10439-016-1678-3.

[172] D. L. Alge, J. Bennett, T. Treasure, S. Voytik-Harbin, W. S. Goebel, and T. M. G. Chu, "Poly(propylene fumarate) reinforced dicalcium phosphate dihydrate cement composites for bone tissue engineering," *Journal of Biomedical Materials Research - Part A*, vol. 100 A, no. 7, pp. 1792–1802, 2012. DOI: 10.1002/jbm.a.34130.

[173] M. G. Gandolfi *et al.*, "Polylactic acid-based porous scaffolds doped with calcium silicate and dicalcium phosphate dihydrate designed for biomedical application," *Materials Science and Engineering C*, vol. 82, no. August 2017, pp. 163–181, 2018. DOI: 10.1016/j.msec.2017.08.040.

[174] L. Jongpaiboonkit, J. W. Halloran, and S. J. Hollister, "Internal structure evaluation of three-dimensional calcium phosphate bone scaffolds: A micro-computed tomographic study," *Journal of the American Ceramic Society*, vol. 89, no. 10, pp. 3176–3181, 2006. DOI: 10.1111/j.1551-2916.2006.01143.x.

[175] G. Li, L. Cuidi, C. Fangping, and L. Changsheng, "Fabrication and characterization of toughness-enhanced scaffolds comprising beta-TCP/POC using the freeform fabrication system with micro-droplet jetting," *Biomedical Materials*, vol. 10, no. 3, 035009 (10 pp.)–035009 (10 pp.) 2015. DOI: 10.1088/1748-6041/10/3/035009.

[176] A. Barba *et al.*, "Osteoinduction by Foamed and 3D-Printed Calcium Phosphate Scaffolds: Effect of Nanostructure and Pore Architecture," *ACS Applied Materials & Interfaces*, acsami.7b14175, 2017. DOI: 10.1021/acsami.7b14175. [Online]. Available: http://pubs.acs.org/doi/10.1021/acsami.7b14175.

[177] O. Suzuki, "Octacalcium phosphate (OCP)-based bone substitute materials," *Japanese Dental Science Review*, vol. 49, no. 2, pp. 58–71, 2013. DOI: 10.1016/j.jdsr.2013.01.001. [Online]. Available: http://dx.doi.org/10.1016/j.jdsr.2013.01.001.

[178] O. Suzuki and T. Anada, "Octacalcium Phosphate: A Potential Scaffold Material for Controlling Activity of Bone-Related Cells In Vitro," *Materials Science Forum*, vol. 783-786, pp. 1366–1371, 2014. DOI: 10.4028/www.scientific.net/MSF.783-786.1366. [Online]. Available: http://www.scientific.net/MSF.783-786.1366.

[179] A. Khalyfa *et al.*, "Development of a new calcium phosphate powder-binder system for the 3D printing of patient specific implants," *Journal of Materials Science: Materials in Medicine*, vol. 18, no. 5, pp. 909–916, 2007. DOI: 10.1007/s10856-006-0073-2.

[180] S. M. Naga, M. Awaad, H. F. El-Maghraby, and A. M. El-Kady, "Biological performance of calcium pyrophosphate-coated porous alumina scaffolds," *International Journal of Applied Ceramic Technology*, vol. 11, no. 1, pp. 1–11, 2014. DOI: 10.1111/ijac.12076.

[181] M. Asadi-Eydivand, M. Solati-Hashjin, S. S. Shafiei, S. Mohammadi, M. Hafezi, and N. A. A. Osman, "Structure, properties, and in vitro behavior of heat-treated calcium sulfate scaffolds fabricated by 3D printing," *PLoS ONE*, vol. 11, no. 3, pp. 1–29, 2016. DOI: 10.1371/journal.pone.0151216.

[182] M. Asadi-Eydivand, M. Solati-Hashjin, A. Farzad, and N. A. Abu Osman, "Effect of technical parameters on porous structure and strength of 3D printed calcium sulfate prototypes," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 57–67, 2016. DOI: 10.1016/j.rcim.2015.06.005. [Online]. Available: http://dx.doi.org/10.1016/j.rcim.2015.06.005.

[183] X. Qi *et al.*, "Three dimensional printing of calcium sulfate and mesoporous bioactive glass scaffolds for improving bone regeneration in vitro and in vivo," *Scientific Reports*, vol. 7, no. November 2016, pp. 2–13, 2017. DOI: 10.1038/srep42556. [Online]. Available: http://dx.doi.org/10.1038/srep42556.

[184] M. Castilho *et al.*, "Application of a 3d printed customized implant for canine cruciate ligament treatment by tibial tuberosity advancement," *Biofabrication*, vol. 6, no. 2, p. 025 005, Mar. 2014. DOI: 10.1088/1758-5082/6/2/025005. [Online]. Available: https://doi.org/10.1088/1758-5082/6/2/025005.

[185] B. Gong, S. Cui, Y. Zhao, Y. Sun, and Q. Ding, "Strain-controlled fatigue behaviors of porous PLA-based scaffolds by 3D-printing technology," *Journal of Biomaterials Science, Polymer Edition*, vol. 28, no. 18, pp. 2196–2204, 2017. DOI: 10.1080/09205063.2017.1388993. [Online]. Available: https://doi.org/10.1080/09205063.2017.1388993.

[186] A. Grémare *et al.*, "Characterization of printed PLA scaffolds for bone tissue engineering," *Journal of Biomedical Materials Research - Part A*, pp. 1–8, 2017. DOI: 10.1002/jbm.a.36289.

[187] Z. Huan, H. Liu, D. Sun, J. Yang, and H. Chu, "Engineered bone scaffolds with dielectrophoresis-based patterning using 3d printing.," *Biomedical Microdevices*, vol. 19, no. 4, 2017. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85034623441%5C&site=eds-live%5C&scope=site.

[188] C. Lam, X. Mo, S. Teoh, and D. Hutmacher, "Scaffold development using 3d printing with a starch-based polymer," *Materials Science and Engineering: C*, vol. 20, no. 1, pp. 49–56, 2002. DOI: https://doi.org/10.1016/S0928-4931(02)00012-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0928493102000127.

[189] H. Liu *et al.*, "Incorporating simvastatin/poloxamer 407 hydrogel into 3d-printed porous ti6al4v scaffolds for the promotion of angiogenesis, osseointegration and bone ingrowth.," *Biofabrication*, vol. 8, no. 4, 2016. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85006052093%5C&site=eds-live%5C&scope=site.

[190] P. Nooeaid *et al.*, "Development of bioactive glass based scaffolds for controlled antibiotic release in bone tissue engineering via biodegradable polymer layered coating.," *Biointerphases*, vol. 9, no. 4, p. 41 001, 2014. DOI: 10.1116/1.4897217. [Online]. Available: http://dx.doi.org/10.1116/1.4897217.

[191] H.-T. Liao, M.-Y. Lee, W.-W. Tsai, H.-C. Wang, and W.-C. Lu, "Osteogenesis of adipose-derived stem cells on polycaprolactone–β-tricalcium phosphate scaffold fabricated via selective laser sintering and surface coating with collagen type I," *Journal of tissue engineering and regenerative medicine*, vol. 4, no. 7, pp. 524–531, 2013. DOI: 10.1002/term. arXiv: NIHMS150003.

[192] E. Mancuso *et al.*, "Three-dimensional printing of porous load-bearing bioceramic scaffolds," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 231, no. 6, pp. 575–585, 2017. DOI: 10.1177/0954411916682984.

[193] D. Hong *et al.*, "Acta Biomaterialia Binder-jetting 3D printing and alloy development of new biodegradable Fe-Mn-Ca / Mg alloys," *Acta Biomaterialia*, vol. 45, pp. 375–386, 2016. DOI: 10.1016/j.actbio.2016.08.032. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2016.08.032.

[194] J. B. Vella, R. P. Trombetta, M. D. Hoffman, J. Inzana, H. Awad, and D. S. W. Benoit, "Three dimensional printed calcium phosphate and poly(caprolactone) composites with improved mechanical properties and preserved microstructure," *Journal of Biomedical Materials Research Part A*, vol. 106, no. 3, pp. 663–672, 2018. DOI: https://doi.org/10.1002/jbm.a.36270. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jbm.a.36270. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jbm.a.36270.

[195] M. Castilho *et al.*, "Fabrication of individual alginate-TCP scaffolds for bone tissue engineering by means of powder printing," *Biofabrication*, vol. 7, no. 1, p. 015 004, Jan. 2015. DOI: 10.1088/1758-5090/7/1/015004. [Online]. Available: https://doi.org/10.1088/1758-5090/7/1/015004.

[196] A. Mazzoli, "Selective laser sintering in biomedical engineering.," *Medical and biological engineering and computing*, vol. 51, no. 3, pp. 245–256, 2013. DOI: 10.1007/s11517-012-1001-x. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23250790.

[197] I. S. Kinstlinger *et al.*, "Open-Source Selective Laser Sintering (OpenSLS) of nylon and biocompatible polycaprolactone," *PLoS ONE*, vol. 11, no. 2, pp. 1–25, 2016. DOI: 10.1371/journal.pone.0147399.

[198] S. Eosoly, D. Brabazon, S. Lohfeld, and L. Looney, "Selective laser sintering of hydroxyapatite/poly-??-caprolactone scaffolds," *Acta Biomaterialia*, vol. 6, no. 7, pp. 2511–2517, 2010. DOI: 10.1016/j.actbio.2009.07.018. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2009.07.018.

[199] A. Gobal and B. Ravani, "Physical Modeling for Selective Laser Sintering (SLS) Process," *Journal of Computing and Information Science in Engineering*, vol. 17, no. c, pp. 1–7, 2016. DOI: 10.1115/1.4034473. [Online]. Available: http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4034473.

[200] B. Partee, S. J. Hollister, and S. Das, "Selective Laser Sintering Process Optimization for Layered Manufacturing of CAPA[sup ®] 6501 Polycaprolactone Bone Tissue Engineering Scaffolds," *Journal of Manufacturing Science and Engineering*, vol. 128, no. May, p. 531, 2006. DOI: 10.1115/1.2162589.

[201] C. Shuai, Z. Mao, H. Lu, Y. Nie, H. Hu, and S. Peng, "Fabrication of porous polyvinyl alcohol scaffold for bone tissue engineering via selective laser sintering.," *Biofabrication*, vol. 5, no. 1, p. 015 014, 2013. DOI: 10.1088/1758-5082/5/1/015014.

[202] Y. Li, D. Li, B. Lu, D. Gao, and J. Zhou, "Current status of additive manufacturing for tissue engineering scaffold," *Rapid Prototyping Journal*, vol. 21, no. 6, pp. 747–762, 2015. DOI: 10.1108/RPJ-03-2014-0029. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-03-2014-0029.

[203] K. V. Wong and A. Hernandez, "A Review of Additive Manufacturing," *ISRN Mechanical Engineering*, vol. 2012, pp. 1–10, 2012. DOI: 10.5402/2012/208760. arXiv: 208760. [Online]. Available: http://www.hindawi.com/journals/isrn.mechanical.engineering/2012/208760/.

[204] A. Mazzoli, C. Ferretti, a. Gigante, E. Salvolini, and M. Mattioli-Belmonte, "Selective laser sintering manufacturing of polycaprolactone bone scaffolds for applications in bone tissue engineering," *Rapid Prototyping Journal*, vol. 21, pp. 386–392, 2015. DOI: 10.1108/RPJ-04-2013-0040. [Online]. Available: http://www.emeraldinsight.com/doi/abs/10.1108/RPJ-04-2013-0040?journalCode=rpj.

[205] C. Shuai *et al.*, "Correlation between properties and microstructure of laser sintered porous β -tricalcium phosphate bone scaffolds," *Science and Technology of Advanced Materials*, vol. 14, no. April, p. 055 002, 2013. DOI: 10.1088/1468-6996/14/5/055002. [Online]. Available: http://stacks.iop.org/1468-6996/14/i=5/a=055002.

[206] J. P. Singh, P. M. Pandey, and A. K. Verma, "Fabrication of three dimensional open porous regular structure of PA-2200 for enhanced strength of scaffold using selective laser sintering," *Rapid Prototyping Journal*, vol. 22, no. 4, pp. 752–765, 2016. DOI: 10.1108/RPJ-11-2014-0148. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-11-2014-0148.

[207] K. H. Tan *et al.*, "Scaffold development using selective laser sintering of polyetheretherketone-hydroxyapatite biocomposite blends," *Biomaterials*, vol. 24, no. 18, pp. 3115–3123, 2003. DOI: 10.1016/S0142-9612(03)00131-5.

[208] K. H. Tan *et al.*, "Selective laser sintering of biocompatible polymers for applications in tissue engineering.," *Bio-medical materials and engineering*, vol. 15, no. 1-2, pp. 113–124, 2005.

[209] P. H. Warnke *et al.*, "Rapid Prototyping: Porous Titanium Alloy Scaffolds Produced by Selective Laser Melting for Bone Tissue Engineering," *Tissue Engineering Part C-Methods*, vol. 15, no. 2, pp. 115–124, 2009. DOI: 10.1089/ten.tec.2008.0288.

[210] Y. Xia *et al.*, "Selective laser sintering fabrication of nano-hydroxyapatite/poly-ϵ-caprolactone scaffolds for bone tissue engineering applications," *International Journal of Nanomedicine*, vol. 8, pp. 4197–4213, 2013. DOI: 10.2147/IJN.S50685. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84887331253%7B%5C&%7DpartnerID=40%7B%5C&%7Dmd5=2ea1b86effd6c59ea9a954d84c82f03c%7B%5C%%7D5Cnhttps://www.dovepress.com/getfile.php?fileID=18030.

[211] W. Y. Zhou, S. H. Lee, M. Wang, W. L. Cheung, and W. Y. Ip, "Selective laser sintering of porous tissue engineering scaffolds from poly(L-lactide)/carbonated hydroxyapatite nanocomposite microspheres," *Journal of Materials Science: Materials in Medicine*, vol. 19, no. 7, pp. 2535–2540, 2008. DOI: 10.1007/s10856-007-3089-3.

[212] B. Duan and M. Wang, "Customized Ca-P/PHBV nanocomposite scaffolds for bone tissue engineering: design, fabrication, surface modification and sustained release of growth factor," *Journal of The Royal Society Interface*, vol. 7, no. Suppl_5, S615–S629, 2010. DOI: 10.1098/rsif.2010.0127.focus. [Online]. Available: http://rsif.royalsocietypublishing.org/cgi/doi/10.1098/rsif.2010.0127.focus.

[213] D. Liu, J. Zhuang, C. Shuai, and S. Peng, "Mechanical properties' improvement of a tricalcium phosphate scaffold with poly-l-lactic acid in selective laser sintering.," *Biofabrication*, vol. 5, no. 2, 025005 (10pp), 2013. DOI: 10.1088/1758-

5082/5/2/025005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23458914.

[214] H. Doyle, S. Lohfeld, and P. McHugh, "Evaluating the effect of increasing ceramic content on the mechanical properties, material microstructure and degradation of selective laser sintered polycaprolactone/??-tricalcium phosphate materials," *Medical Engineering and Physics*, vol. 37, no. 8, pp. 767–776, 2015. DOI: 10.1016/j.medengphy.2015.05.009. [Online]. Available: http://dx.doi.org/10.1016/j.medengphy.2015.05.009.

[215] H. Doyle, S. Lohfeld, and P. McHugh, "Predicting the elastic properties of selective laser sintered PCL/β-TCP bone scaffold materials using computational modelling," *Annals of Biomedical Engineering*, vol. 42, no. 3, pp. 661–677, Sep. 2013. DOI: 10.1007/s10439-013-0913-4. [Online]. Available: https://doi.org/10.1007/s10439-013-0913-4.

[216] Y. Du, H. Liu, J. Shuang, J. Wang, J. Ma, and S. Zhang, "Microsphere-based selective laser sintering for building macroporous bone scaffolds with controlled microstructure and excellent biocompatibility," *Colloids and Surfaces B: Biointerfaces*, vol. 135, pp. 81–89, 2015. DOI: 10.1016/j.colsurfb.2015.06.074.

[217] Y. Du *et al.*, "Selective laser sintering scaffold with hierarchical architecture and gradient composition for osteochondral repair in rabbits," *Biomaterials*, vol. 137, pp. 37–48, 2017. DOI: 10.1016/j.biomaterials.2017.05.021. [Online]. Available: http://dx.doi.org/10.1016/j.biomaterials.2017.05.021.

[218] S. Eshraghi and S. Das, "Mechanical and microstructural properties of polycaprolactone scaffolds with one-dimensional, two-dimensional, and three-dimensional orthogonally oriented porous architectures produced by selective laser sintering," *Acta Biomaterialia*, vol. 6, no. 7, pp. 2467–2476, 2010. DOI: 10.1016/j.actbio.2010.02.002. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2010.02.002.

[219] S. Eshraghi and S. Das, "Micromechanical finite-element modeling and experimental characterization of the compressive mechanical properties of polycaprolactone-hydroxyapatite composite scaffolds prepared by selective laser sintering for bone tissue engineering," *Acta Biomaterialia*, vol. 8, no. 8, pp. 3138–3143, 2012. DOI: 10.1016/j.actbio.2012.04.022. arXiv: NIHMS150003. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2012.04.022.

[220] T. Pereira *et al.*, "Effect of process parameters on the properties of selective laser sintered Poly(3-hydroxybutyrate) scaffolds for bone tissue engineering," *Virtual and Physical Prototyping*, vol. 7, no. 4, pp. 275–285, 2012. DOI: 10.1080/17452759.2012.738551. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/17452759.2012.738551.

[221] A. Bandyopadhyay, S. Bose, and S. Das, "3D printing of biomaterials," *MRS Bulletin*, vol. 40, no. 2, pp. 108–114, 2015. DOI: 10.1557/mrs.2015.3.

[222] G. V. Salmoria, R. V. Pereira, M. C. Fredel, and A. P. Casadei, "Properties of PLDLA/bioglass scaffolds produced by selective laser sintering," *Polymer Bulletin*, vol. 75, no. 3, pp. 1299–1309, 2018. DOI: 10.1007/s00289-017-2093-0.

[223] C. Shuai, J. Zhuang, S. Peng, and X. Wen, "Inhibition of phase transformation from *beta-* to α-tricalcium phosphate with addition of poly (L-lactic acid) in selective laser sintering," *Rapid Prototyping Journal*, vol. 20, no. 5, pp. 369–376, 2014. DOI: 10.1108/RPJ-03-2013-0037. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-03-2013-0037.

[224] W. Y. Zhou, M. Wang, W. L. Cheung, and W. Y. Ip, "Selective Laser Sintering of Poly (L-Lactide)/ Carbonated Hydroxyapatite Nanocomposite Porous Scaffolds for Bone Tissue Engineering," *Tissue Engineering*, pp. 179–204, 2010. DOI: 10.5772/8574.

[225] T. Kumaresan, R. Gandhinathan, M. Ramu, M. Ananthasubramanian, and K. B. Pradheepa, "Design, analysis and fabrication of polyamide/ hydroxyapatite porous structured scaffold using selective laser sintering method for biomedical applications," *Journal of Mechanical Science and Technology*, vol. 30, no. 11, pp. 5305–5312, 2016. DOI: 10.1007/s12206-016-1049-x. [Online]. Available: http://link.springer.com/10.1007/s12206-016-1049-x.

[226] M. Roskies *et al.*, "Improving PEEK bioactivity for craniofacial reconstruction using a 3D printed scaffold embedded with mesenchymal stem cells," *Journal of Biomaterials Applications*, vol. 31, no. 1, pp. 132–139, 2016. DOI: 10.1177/0885328216638636.

[227] K. C. R. Kolan, M. C. Leu, G. E. Hilmas, R. F. Brown, and M. Velez, "Fabrication of 13-93 bioactive glass scaffolds for bone tissue engineering using indirect selective laser sintering," *Biofabrication*, vol. 3, no. 2, 2011. DOI: 10.1088/1758-5082/3/2/025004.

[228] K. C. R. Kolan, A. Thomas, M. C. Leu, and G. Hilmas, "In vitro assessment of laser sintered bioactive glass scaffolds with different pore geometries," *Rapid Prototyping Journal*, vol. 21, no. 2, pp. 152–158, 2015. DOI: 10.1108/RPJ-12-2014-0175. arXiv: arXiv:1011.1669v3. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-12-2014-0175.

[229] R. d. V. Pereira, G. V. Salmoria, M. O. C. de Moura, Á. Aragones, and M. C. Fredel, "Scaffolds of PDLLA/bioglass 58S produced via selective laser sintering," *Materials Research*, vol. 17, no. suppl 1, pp. 33–38, 2014. DOI: 10.1590/S1516-14392014005000075. [Online]. Available: http://www.scielo.br/scielo.php?script=sci%7B%5C_%7Darttext%7B%5C&%7Dpid=S1516-14392014000700007%7B%5C&%7Dlng=en%7B%5C&%7Dtlng=en.

[230] R. Jackson *et al.*, "Chemically Treated 3D Printed Polymer Scaffolds for Biomineral Formation," *ACS Omega*, vol. 3, no. 4, 2018. DOI: 10.1021/acsomega.8b00219.

[231] M. Grau *et al.*, "In Vitro Evaluation of PCL and P(3HB) as Coating Materials for Selective Laser Melted Porous Titanium Implants," *Materials*, vol. 10, no. 12, p. 1344, 2017. DOI: 10.3390/ma10121344. [Online]. Available: http://www.mdpi.com/1996-1944/10/12/1344.

[232] P. K. Srivas *et al.*, "Osseointegration assessment of extrusion printed Ti6Al4V scaffold towards accelerated skeletal defect healing via tissue in-growth," *Bioprinting*, vol. 6, no. January, pp. 8–17, 2017. DOI: 10.1016/j.bprint.2017.04.002. [Online]. Available: http://dx.doi.org/10.1016/j.bprint.2017.04.002.

[233] X. Ye, S. Leeflang, C. Wu, J. Chang, J. Zhou, and Z. Huan, "Mesoporous bioactive glass functionalized 3D Ti-6Al-4V Scaffolds with improved surface bioactivity," *Materials*, vol. 10, no. 11, 2017. DOI: 10.3390/ma10111244.

[234] A. Jonitz-Heincke, J. Wieding, C. Schulze, D. Hansmann, and R. Bader, "Comparative analysis of the oxygen supply and viability of human osteoblasts in three-dimensional titanium scaffolds produced by laser-beam or electron-beam melting," *Materials*, vol. 6, no. 11, pp. 5398–5409, 2013. DOI: 10.3390/ma6115398.

[235] J. Markhoff, J. Wieding, V. Weissmann, J. Pasold, A. Jonitz-Heincke, and R. Bader, "Influence of Different Three-Dimensional Open Porous Titanium Scaffold Designs on Human Osteoblasts Behavior in Static and Dynamic Cell Investigations," *Materials*, vol. 8, no. 8, pp. 5490–5507, 2015. DOI: 10.3390/ma8085259. [Online]. Available: http://www.mdpi.com/1996-1944/8/8/5259/htm.

[236] A. Azidin *et al.*, "Investigation of mechanical properties for open cellular structure CoCrMo alloy fabricated by selective laser melting process," *IOP Conference Series: Materials Science and Engineering*, vol. 100, p. 012 033, 2015. DOI: 10.1088/1757-899X/100/1/012033. [Online]. Available: http://stacks.iop.org/1757-899X/100/i=1/a=012033?key=crossref.211f9080ae03ca17a3089770a49caf65.

[237] D. Joguet, S. Costil, H. Liao, and Y. Danlos, "Porosity content control of CoCrMo and titanium parts by Taguchi method applied to selective laser melting process parameter," *Rapid Prototyping Journal*, vol. 22, no. 1, pp. 20–30, 2016. DOI: 10.1108/RPJ-09-2013-0092. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/RPJ-09-2013-0092.

[238] Y. Li *et al.*, "Additively manufactured biodegradable porous magnesium," *Acta Biomaterialia*, 2017. DOI: 10.1016/j.actbio.2017.12.008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S174270611730764X.

[239] J. Korpela, A. Kokkari, H. Korhonen, M. Malin, T. Narhi, and J. Seppalea, "Biodegradable and bioactive porous scaffold structures prepared using fused deposition modeling," *Journal of Biomedical Materials Research - Part B Applied Biomaterials*, vol. 101, no. 4, pp. 610–619, 2013. DOI: 10.1002/jbm.b.32863.

[240] R. Smeets *et al.*, "Selective laser-melted fully biodegradable scaffold composed of poly(d,l-lactide) and ??-tricalcium phosphate with potential as a biodegradable implant for complex maxillofacial reconstruction: In vitro and in vivo results," *Journal of Biomedical Materials Research - Part B Applied Biomaterials*, vol. 105, no. 5, pp. 1–16, 2016. DOI: 10.1002/jbm.b.33660.

[241] K. Szlazak *et al.*, "Characterization of three-dimensional printed composite scaffolds prepared with different fabrication methods," *Archives of Metallurgy and Materials*, vol. 61, no. 2A, pp. 645–649, 2016. DOI: 10.1515/amm-2016-0110.

[242] H. Zhang *et al.*, "Three dimensional printed polylactic acid-hydroxyapatite composite scaffolds for prefabricating vascularized tissue engineered bone: An in vivo bioreactor model," *Scientific Reports*, vol. 7, no. 1, pp. 1–13, 2017. DOI: 10.1038/s41598-017-14923-7.

[243] C. Shuai, Y. Li, P. Feng, W. Guo, W. Yang, and S. Peng, "Positive feedback effects of Mg on the hydrolysis of poly-l-lactic acid (PLLA): Promoted degradation of PLLA scaffolds," *Polymer Testing*, vol. 68, no. March, pp. 27–33, 2018. DOI: 10.1016/j.polymertesting.2018.03.042. [Online]. Available: https://doi.org/10.1016/j.polymertesting.2018.03.042.

[244] H. Zhang, J. Liang, Y. Ding, and P. Li, "The controlled release of growth factor via modified coaxial electrospun fibres with emulsion or hydrogel as the core," *Materials Letters*, vol. 181, pp. 119–122, 2016. DOI: 10.1016/j.matlet.2016.05.146.

[245] S. Barui, S. Chatterjee, S. Mandal, A. Kumar, and B. Basu, "Microstructure and compression properties of 3D powder printed Ti-6Al-4V scaffolds with designed porosity: Experimental and computational analysis," *Materials Science and Engineering C*, vol. 70, pp. 812–823, 2017. DOI: 10.1016/j.msec.2016.09.040. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2016.09.040.

[246] J. An, J. E. M. Teoh, R. Suntornnond, and C. K. Chua, "Design and 3D Printing of Scaffolds and Tissues," *Engineering*, vol. 1, no. 2, pp. 261–268, 2015. DOI: 10.15302/J-ENG-2015061. [Online]. Available: http://dx.doi.org/10.15302/J-ENG-2015061.

[247] K. A. Mumtaz and N. Hopkinson, "Laser melting functionally graded composition of Waspaloy?? and Zirconia powders," *Journal of Materials Science*, vol. 42, no. 18, pp. 7647–7656, 2007. DOI: 10.1007/s10853-007-1661-3.

[248] C. C. Ng, M. Savalani, and H. C. Man, "Fabrication of magnesium using selective laser melting technique," *Rapid Prototyping Journal*, vol. 17, no. 6, pp. 479–490, 2011. DOI: 10.1108/13552541111184206.

[249] A. G. Demir, L. Monguzzi, and B. Previtali, "Selective laser melting of pure Zn with high density for biodegradable implant manufacturing," *Additive Manufacturing*, vol. 15, pp. 20–28, 2017. DOI: http://dx.doi.org/10.1016/j.addma.2017.03.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214860416301555.

[250] M. Lindner *et al.*, "Manufacturing of individual biodegradable bone substitute implants using selective laser melting technique," *Journal of Biomedical Materials Research - Part A*, vol. 97 A, no. 4, pp. 466–471, 2011. DOI: 10.1002/jbm.a.33058.

[251] M. M. Savalani and J. M. Pizarro, "Effect of preheat and layer thickness on selective laser melting (SLM) of magnesium," *Rapid Prototyping Journal*, vol. 22, no. 1, pp. 115–122, 2016. DOI: 10.1108/RPJ-07-2013-0076. [Online]. Available: http://dx.doi.org/10.1108/RPJ-07-2013-0076.

[252] M. Savalani, C. Ng, and H. Man, "Selective Laser Melting of Magnesium for Future Applications in Medicine," *2010 International Conference on Manufacturing Automation*, pp. 50–54, 2010. DOI: 10.1109/ICMA.2010.45.

[253] N. W. Hrabe, P. Heinl, R. K. Bordia, C. Körner, and R. J. Fernandes, "Maintenance of a bone collagen phenotype by osteoblast-like cells in 3D periodic porous titanium (Ti-6Al-4 V) structures fabricated by selective electron beam melting.," *Connective tissue research*, vol. 54, no. 6, pp. 351–60, 2013. DOI: 10.3109/03008207.2013.822864. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3864771%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract.

[254] Z. Jia *et al.*, "A novel cytocompatible, hierarchical porous Ti6Al4V scaffold with immobilized silver nanoparticles," *Materials Letters*, vol. 157, pp. 143–146, 2015. DOI: 10.1016/j.matlet.2015.05.084. [Online]. Available: http://dx.doi.org/10.1016/j.matlet.2015.05.084.

[255] J. Lv, P. Xiu, J. Tan, Z. Jia, H. Cai, and Z. Liu, "Enhanced angiogenesis and osteogenesis in critical bone defects by the controlled release of BMP-2 and VEGF: implantation of electron beam melting-fabricated porous Ti 6 Al 4 V scaffolds incorporating growth factor-doped fibrin glue," *Biomedical Materials*, vol. 10, no. 3, p. 035 013, 2015. DOI: 10.1088/1748-6041/10/3/035013. [Online]. Available: http://iopscience.iop.org/1748-605X/10/3/035013/article/%7B%5C%%7D5Cnhttp://stacks.iop.org/1748-605X/10/i=3/a=035013?key=crossref.be532a51be12839084250853af7d1ab8.

[256] G. M. de Peppo *et al.*, "Free-form-fabricated commercially pure Ti and Ti6Al4V porous scaffolds support the growth of human embryonic stem cell-derived mesodermal progenitors.," *TheScientificWorldJournal*, vol. 2012, p. 646 417, 2012. DOI: 10.1100/2012/646417. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84856401819%7B%5C&%7DpartnerID=tZOtx3y1.

[257] A. Rodriguez-Palomo, D. Monopoli, H. Afonso, I. Izquierdo-Barba, and M. Vallet-Regí, "Surface zwitterionization of customized 3D Ti6Al4V scaffolds: a promising alternative to eradicate bone infection," *J. Mater. Chem. B*, pp. 4356–4365, 2016. DOI: 10.1039/C6TB00675B. [Online]. Available: http://xlink.rsc.org/?DOI=C6TB00675B.

[258] S. Naghieh, M. R. Karamooz Ravari, M. Badrossamay, E. Foroozmehr, and M. Kadkhodaei, "Numerical investigation of the mechanical properties of the additive manufactured bone scaffolds fabricated by FDM: The effect of layer penetration and post-heating," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 59, pp. 241–250, 2016. DOI: 10.1016/j.jmbbm.2016.01.031. [Online]. Available: http://dx.doi.org/10.1016/j.jmbbm.2016.01.031.

[259] M. Cavo and S. Scaglione, "Scaffold microstructure effects on functional and mechanical performance: Integration of theoretical and experimental approaches for bone tissue engineering applications," *Materials Science and Engineering C*, vol. 68, pp. 872–879, 2016. DOI: 10.1016/j.msec.2016.07.041. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2016.07.041.

[260] K. M. Ferlin, M. E. Prendergast, M. L. Miller, D. S. Kaplan, and J. P. Fisher, "Influence of 3D printed porous architecture on mesenchymal stem cell enrichment and differentiation," *Acta Biomaterialia*, vol. 32, pp. 161–169, 2016. DOI: 10.1016/j.actbio.2016.01.007. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2016.01.007.

[261] D. Steffens *et al.*, "3d-printed PCL scaffolds for the cultivation of mesenchymal stem cells," *Journal of Applied Biomaterials and Functional Materials*, vol. 14, no. 1, pp. 19–25, Apr. 2016. DOI: 10.5301/jabfm.5000252. [Online]. Available: https://doi.org/10.5301/jabfm.5000252.

[262] G. Vozzi, a. Previti, D. De Rossi, and a. Ahluwalia, "Microsyringe-based deposition of two-dimensional and three-dimensional polymer scaffolds with a well-defined geometry for application to tissue engineering.," *Tissue engineering*, vol. 8, no. 6, pp. 1089–1098, 2002. DOI: 10.1089/107632702320934182.

[263] M. A. Nowicki, N. J. Castro, M. W. Plesniak, and L. G. Zhang, "3d printing of novel osteochondral scaffolds with graded microstructure," *Nanotechnology*, vol. 27, no. 41, p. 414 001, Sep. 2016. DOI: 10.1088/0957-4484/27/41/414001. [Online]. Available: https://doi.org/10.1088/0957-4484/27/41/414001.

[264] A. Nommeots-Nomm, P. D. Lee, and J. R. Jones, "Direct ink writing of highly bioactive glasses," *Journal of the European Ceramic Society*, vol. 38, no. 3, pp. 837–844, 2018. DOI: 10.1016/j.jeurceramsoc.2017.08.006. [Online]. Available: https://doi.org/10.1016/j.jeurceramsoc.2017.08.006.

[265] Y. Yan, R. Wu, R. Zhang, Z. Xiong, and F. Lin, "Biomaterial forming research using RP technology," *Rapid Prototyping Journal*, vol. 9, no. 3, pp. 142–149, 2003. DOI: 10.1108/13552540310477445.

[266] C. Mota *et al.*, "Additive manufacturing of poly[(R)-3-hydroxybutyrate-co-(R)-3-hydroxyhexanoate] scaffolds for engineered bone development," *Journal of Tissue Engineering and Regenerative Medicine*, vol. 11, no. 1, pp. 175–186, 2017. DOI: 10.1002/term.1897.

[267] D. Puppi *et al.*, "Additive manufacturing of wet-spun polymeric scaffolds for bone tissue engineering," *Biomedical Microdevices*, vol. 14, no. 6, pp. 1115–1127, 2012. DOI: 10.1007/s10544-012-9677-0.

[268] D. Puppi, A. M. Piras, A. Pirosa, S. Sandreschi, and F. Chiellini, "Levofloxacin-loaded star poly(ϵ-caprolactone) scaffolds by additive manufacturing," *Journal of Materials Science: Materials in Medicine*, vol. 27, no. 3, pp. 1–11, 2016. DOI: 10.1007/s10856-015-5658-1.

[269] E. Malikmammadov, T. E. Tanir, A. Kiziltay, V. Hasirci, and N. Hasirci, "PCL-TCP wet spun scaffolds carrying antibiotic-loaded microspheres for bone tissue engineering," *Journal of Biomaterials Science, Polymer Edition*, vol. 29, no. 7-9, pp. 805–824, 2018. DOI: 10.1080/09205063.2017.1354671. [Online]. Available: https://doi.org/10.1080/09205063.2017.1354671.

[270] S. Ucar, M. Ermis, and N. Hasirci, "Modified chitosan scaffolds: Proliferative, cytotoxic, apoptotic, and necrotic effects on Saos-2 cells and antimicrobial effect on Escherichia coli," *Journal of Bioactive and Compatible Polymers*, vol. 31, no. 3, pp. 304–319, 2015. DOI: 10.1177/0883911515627471.

[271] M. T. Rodrigues *et al.*, "Bone marrow stromal cells on a three-dimensional bioactive fiber mesh undergo osteogenic differentiation in the absence of osteogenic media supplements: The effect of silanol groups," *Acta Biomaterialia*, vol. 10, no. 10, pp. 4175–4185, 2014. DOI: 10.1016/j.actbio.2014.05.026.

[272] F. Dini *et al.*, "Tailored star poly (ϵ-caprolactone) wet-spun scaffolds for in vivo regeneration of long bone critical size defects," *Journal of Bioactive and Compatible Polymers*, vol. 31, no. 1, pp. 15–30, 2016. DOI: 10.1177/0883911515597928.

[273] C. Mota, D. Puppi, D. Dinucci, M. Gazzarri, and F. Chiellini, "Additive manufacturing of star poly(*epsilon*-caprolactone) wet-spun scaffolds for bone tissue engineering applications," *Journal of Bioactive and Compatible Polymers*, vol. 28, no. 4, pp. 320–340, 2013. DOI: 10.1177/0883911513490341.

[274] D. Puppi *et al.*, "Design, fabrication and characterization of tailored poly[(r)-3-hydroxybutyrate-co-(r)-3-hydroxyexanoate] scaffolds by computer-aided wet-spinning.," *Rapid Prototyping Journal*, vol. 24, no. 1, pp. 1–8, 2018. DOI: 10.1108/RPJ-03-2016-0037. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-85041816994%5C&site=eds-live%5C&scope=site.

[275] S. Eqtesadi, A. Motealleh, P. Miranda, A. Pajares, A. Lemos, and J. M. Ferreira, "Robocasting of 45S5 bioactive glass scaffolds for bone tissue engineering," *Journal of the European Ceramic Society*, vol. 34, no. 1, pp. 107–118, 2014. DOI: 10.1016/j.jeurceramsoc.2013.08.003. [Online]. Available: http://dx.doi.org/10.1016/j.jeurceramsoc.2013.08.003.

[276] C. F. Marques *et al.*, "Biphasic calcium phosphate scaffolds fabricated by direct write assembly: Mechanical, anti-microbial and osteoblastic properties," *Journal of the European Ceramic Society*, vol. 37, no. 1, pp. 359–368, 2017. DOI: 10.1016/j.jeurceramsoc.2016.08.018. [Online]. Available: http://dx.doi.org/10.1016/j.jeurceramsoc.2016.08.018.

[277] P. Miranda, A. Pajares, E. Saiz, A. P. Tomsia, and F. Guiberteau, "Mechanical properties of calcium phosphate scaffolds fabricated by robocasting," *Journal of Biomedical Materials Research - Part A*, vol. 85, no. 1, pp. 218–227, 2008. DOI: 10.1002/jbm.a.31587.

[278] P. Miranda, E. Saiz, K. Gryn, and A. P. Tomsia, "Sintering and robocasting of β-tricalcium phosphate scaffolds for orthopaedic applications," *Acta Biomaterialia*, vol. 2, no. 4, pp. 457–466, 2006. DOI: 10.1016/j.actbio.2006.02.004.

[279] P. D. Dalton, C. Vaquette, B. L. Farrugia, T. R. Dargaville, T. D. Brown, and D. W. Hutmacher, "Electrospinning and additive manufacturing: Converging technologies," *Biomater. Sci.*, vol. 1, pp. 171–185, 2 2013. DOI: 10.1039/C2BM00039C. [Online]. Available: http://dx.doi.org/10.1039/C2BM00039C.

[280] S. M. Giannitelli, P. Mozetic, M. Trombetta, and A. Rainer, "Combined additive manufacturing approaches in tissue engineering," *Acta Biomaterialia*, vol. 24, pp. 1–11, 2015. DOI: 10.1016/j.actbio.2015.06.032. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2015.06.032.

[281] L. Elomaa, C.-C. Pan, Y. Shanjani, A. Malkovskiy, J. V. Seppälä, and Y. Yang, "Three-dimensional fabrication of cell-laden biodegradable poly(ethylene glycol-co-depsipeptide) hydrogels by visible light stereolithography," *J. Mater. Chem. B*, vol. 3, no. 42, pp. 8348–8358, 2015. DOI: 10.1039/C5TB01468A. [Online]. Available: http://xlink.rsc.org/?DOI=C5TB01468A.

[282] B. van Bochove, G. Hannink, P. Buma, and D. W. Grijpma, "Preparation of Designed Poly(trimethylene carbonate) Meniscus Implants by Stereolithography: Challenges in Stereolithography," *Macromolecular Bioscience*, pp. 1853–1863, 2016. DOI: 10.1002/mabi.201600290.

[283] I. Izquierdo-Barba, "Scaffold Designing," *Bio-Ceramics with Clinical Applications*, vol. 9781118406, pp. 291–313, 2014. DOI: 10.1002/9781118406748.ch10.

[284] O. Guillaume *et al.*, "Poly(trimethylene carbonate) and nano-hydroxyapatite porous scaffolds manufactured by stereolithography," *Polymers for Advanced Technologies*, vol. 28, no. 10, pp. 1219–1225, 2017. DOI: 10.1002/pat.3892.

[285] F. Scalera, C. Esposito Corcione, F. Montagna, A. Sannino, and A. Maffezzoli, "Development and characterization of UV curable epoxy/hydroxyapatite suspensions for stereolithography applied to bone tissue engineering," *Ceramics International*, vol. 40, no. 10, pp. 15 455–15 462, 2014. DOI: 10.1016/j.ceramint.2014.06.117. [Online]. Available: http://dx.doi.org/10.1016/j.ceramint.2014.06.117.

[286] A. Singh and A. K. Gaharwar, "Microscale technologies for cell engineering," *Microscale Technologies for Cell Engineering*, pp. 1–318, 2015. DOI: 10.1007/978-3-319-20726-1.

[287] D. Dean *et al.*, "Multiple initiators and dyes for continuous Digital Light Processing (cDLP) additive manufacture of resorbable bone tissue engineering scaffolds," *Virtual and Physical Prototyping*, vol. 9, no. 1, pp. 3–9, 2014. DOI: 10.1080/17452759.2013.873337. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/17452759.2013.873337.

[288] P. Tesavibul *et al.*, "Biocompatibility of hydroxyapatite scaffolds processed by lithography-based additive manufacturing," *Bio-Medical Materials and Engineering*, vol. 26, no. 1-2, pp. 31–38, 2015. DOI: 10.3233/BME-151549.

[289] J. Wallace *et al.*, "Validating continuous digital light processing (cDLP) additive manufacturing accuracy and tissue engineering utility of a dye-initiator package.," *Biofabrication*, vol. 6, no. 1, p. 015 003, 2014. DOI: 10.1088/1758-5082/6/1/015003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24429508.

[290] D. Dean *et al.*, "Continuous digital light processing (cDLP): Highly accurate additive manufacturing of tissue engineered bone scaffolds," *Virtual and Physical Prototyping*, vol. 7, no. 1, pp. 13–24, 2012. DOI: 10.1080/17452759.2012.673152. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/17452759.2012.673152.

[291] R. J. Mondschein, A. Kanitkar, C. B. Williams, S. S. Verbridge, and T. E. Long, "Polymer structure-property requirements for stereolithographic 3D printing of soft tissue engineering scaffolds," *Biomaterials*, vol. 140, pp. 170–188, 2017. DOI: 10.1016/j.biomaterials.2017.06.005. [Online]. Available: http://dx.doi.org/10.1016/j.biomaterials.2017.06.005.

[292] M. Dadsetan *et al.*, "Effect of calcium phosphate coating and rhBMP-2 on bone regeneration in rabbit calvaria using poly(propylene fumarate) scaffolds," *Acta Biomaterialia*, vol. 18, pp. 9–20, 2015. DOI: 10.1016/j.actbio.2014.12.024. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2014.12.024.

[293] F. P. Melchels, J. Feijen, and D. W. Grijpma, "A review on stereolithography and its applications in biomedical engineering," *Biomaterials*, vol. 31, no. 24, pp. 6121–6130, 2010. DOI: 10.1016/j.biomaterials.2010.04.050. [Online]. Available: http://dx.doi.org/10.1016/j.biomaterials.2010.04.050.

[294] B. Thavornyutikarn *et al.*, "Porous 45S5 Bioglass®-based scaffolds using stereolithography: Effect of partial pre-sintering on structural and mechanical properties of scaffolds," *Materials Science and Engineering C*, vol. 75, pp. 1281–1288, 2017. DOI: 10.1016/j.msec.2017.03.001. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2017.03.001.

[295] J. Yao, C. Bastiaansen, and T. Peijs, "High Strength and High Modulus Electrospun Nanofibers," *Fibers*, vol. 2, no. 2, pp. 158–186, 2014. DOI: 10.3390/fib2020158. [Online]. Available: http://www.mdpi.com/2079-6439/2/2/158/.

[296] S. Comlekci *et al.*, "Electrostatic field considerations related force effect on electrospinning," *IEEE Transactions on Nanotechnology*, vol. 103, no. 1, pp. 149–151, 2013. DOI: 10.1063/1.2924439. [Online]. Available: http://dx.doi.org/10.1016/j.elstat.2011.12.001.

[297] O. Karatay, "Modelling of electrospinning process at various electric fields," English, *Micro & Nano Letters*, vol. 6, 858–862(4), 10 Oct. 2011. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/mnl.2011.0440.

[298] O. Karatay, M. Doğan, T. Uyar, D. Çökeliler, and İ. C. Koçum, "An alternative electrospinning approach with varying electric field for 2-d-aligned nanofibers," *IEEE Transactions on Nanotechnology*, vol. 13, no. 1, pp. 101–108, 2014. DOI: 10.1109/TNANO.2013.2293704.

[299] A. R. D'Amato *et al.*, "Evaluation of procedures to quantify solvent retention in electrospun fibers and facilitate solvent removal," *Fibers and Polymers*, vol. 18, no. 3, pp. 483–492, 2017. DOI: 10.1007/s12221-017-1061-5. [Online]. Available: http://link.springer.com/10.1007/s12221-017-1061-5.

[300] S. H. Park, T. G. Kim, H. C. Kim, D. Y. Yang, and T. G. Park, "Development of dual scale scaffolds via direct polymer melt deposition and electrospinning for applications in tissue regeneration," *Acta Biomaterialia*, vol. 4, no. 5, pp. 1198–1207, 2008. DOI: 10.1016/j.actbio.2008.03.019.

[301] J. Venugopal, P. Vadgama, T. S. S. Kumar, and S. Ramakrishna, "Biocomposite nanofibres and osteoblasts for bone tissue engineering," *Nanotechnology*, vol. 18, no. 5, p. 055 101, 2007. DOI: 10.1088/0957-4484/18/5/055101. [Online]. Available: http://dx.doi.org/10.1088/0957-4484/18/5/055101%7B%5C%%7D5Cnhttp://stacks.iop.org/0957-4484/18/i=5/a=055101?key=crossref.253d9101fbf8debfcf9ef468ac492076.

[302] M. I. Hassan, N. Sultana, and S. Hamdan, "Bioactivity Assessment of Poly(ϵ-caprolactone)/Hydroxyapatite Electrospun Fibers for Bone Tissue Engineering Application," *Journal of Nanomaterials*, vol. 2014, pp. 1–6, 2014. DOI: 10.1155/2014/573238.

[303] V. Guarino, M. Raucci, A. Ronca, V. Cirillo, and L. Ambrosio, *5 – Multifunctional scaffolds for bone regeneration*. Woodhead Publishing Limited, 2014, pp. 95–117, ISBN: 9780857094971. DOI: 10.1533/9780857099037.2.95. [Online]. Available: http://dx.doi.org/10.1533/9780857099037.2.95.

[304] M. Komárek and L. Martinová, "Design and evaluation of melt-electrospinning electrodes," *Proceedings of the 2nd Nanocon*, Jan. 2010.

[305] H. Zhou, T. B. Green, and Y. L. Joo, "The thermal effects on electrospinning of polylactic acid melts," *Polymer*, vol. 47, no. 21, pp. 7497–7505, 2006. DOI: https://doi.org/10.1016/j.polymer.2006.08.042. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0032386106009955.

[306] E. Zhmayev, H. Zhou, and Y. L. Joo, "Modeling of non-isothermal polymer jets in melt electrospinning," *Journal of Non-Newtonian Fluid Mechanics*, vol. 153, no. 2-3, pp. 95–108, 2008. DOI: 10.1016/j.jnnfm.2007.11.011.

[307] G. Hochleitner *et al.*, "Additive manufacturing of scaffolds with sub-micron filaments via melt electrospinning writing.," *Biofabrication*, vol. 7, no. 3, p. 035 002, 2015. DOI: 10.1088/1758-5090/7/3/035002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/26065373.

[308] Y. Yang *et al.*, "Effect of electric field distribution uniformity on electrospinning," *Journal of Applied Physics*, vol. 103, no. 10, pp. 1–12, 2008. DOI: 10.1063/1.2924439.

[309] K. (1.) Auyson, P. (1.) Koomsap, A. (1.) Chanthakulchan, and P. (2.) Supaphol, "Investigation of applying electrospinning in fused deposition modeling for scaffold fabrication.," in *High Value Manufacturing: Advanced Research in Virtual and Rapid Prototyping - Proceedings of the 6th International Conference on Advanced Research and Rapid Prototyping, VRP 2013*, (1)Asian Institute of Technology, 2014, pp. 149–154. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edselc%5C&AN=edselc.2-52.0-84892178126%5C&site=eds-live%5C&scope=site.

[310] S. Mohammadzadehmoghadam, Y. Dong, and I. J. Davies, "Modeling electrospun nanofibers: An overview from theoretical, empirical, and numerical approaches," *International Journal of Polymeric Materials and Polymeric Biomaterials*, vol. 65, no. 17, pp. 901–915, 2016. DOI: 10.1080/00914037.2016.1180617.

[311] C. J. Thompson, G. G. Chase, A. L. Yarin, and D. H. Reneker, "Effects of parameters on nanofiber diameter determined from electrospinning model," *Polymer*, vol. 48, no. 23, pp. 6913–6922, 2007. DOI: 10.1016/j.polymer.2007.09.017. [Online]. Available: http://dx.doi.org/10.1016/j.polymer.2007.09.017.

[312] B. H. Moghadam, A. K. Haghi, S. Kasaei, and M. Hasanzadeh, "Computational-based approach for predicting porosity of electrospun nanofiber mats using response surface methodology and artificial neural network methods," *Journal of Macromolecular Science, Part B*, vol. 54, no. 11, pp. 1404–1425, 2015. DOI: 10.1080/00222348.2015.1090654. eprint: https://doi.org/10.1080/00222348.2015.1090654. [Online]. Available: https://doi.org/10.1080/00222348.2015.1090654.

[313] E. Zhmayev, D. Cho, and Y. L. Joo, "Modeling of melt electrospinning for semi-crystalline polymers," *Polymer*, vol. 51, no. 1, pp. 274–290, 2010. DOI: 10.1016/j.polymer.2009.11.025. [Online]. Available: http://dx.doi.org/10.1016/j.polymer.2009.11.025.

[314] A. L. Yarin, S. Koombhongse, and D. H. Reneker, "Taylor cone and jetting from liquid droplets in electrospinning of nanofibers," *Journal of Applied Physics*, vol. 90, no. 9, pp. 4836–4846, 2001. DOI: 10.1063/1.1408260. eprint: https://doi.org/10.1063/1.1408260. [Online]. Available: https://doi.org/10.1063/1.1408260.

[315] D. H. Reneker, A. L. Yarin, H. Fong, and S. Koombhongse, "Bending instability of electrically charged liquid jets of polymer solutions in electrospinning," *Journal of Applied Physics*, vol. 87, no. 9, pp. 4531–4547, 2000. DOI: 10.1063/1.373532. eprint: https://doi.org/10.1063/1.373532. [Online]. Available: https://doi.org/10.1063/1.373532.

[316] G. Hochleitner, M. Kessler, M. Schmitz, A. R. Boccaccini, J. Teβmar, and J. Groll, "Melt electrospinning writing of defined scaffolds using polylactide-poly(ethylene glycol) blends with 45S5 bioactive glass particles," *Materials Letters*, vol. 205, pp. 257–260, 2017. DOI: 10.1016/j.matlet.2017.06.096.

[317] J. Gilmore, T. Burg, R. E. Groff, and K. J. Burg, "Design and optimization of a novel bio-loom to weave melt-spun absorbable polymers for bone tissue engineering," *Journal of Biomedical Materials Research - Part B Applied Biomaterials*, vol. 105, no. 6, pp. 1342–1351, 2017. DOI: 10.1002/jbm.b.33700.

[318] M. L. Muerza-Cascante *et al.*, "Endosteal-like extracellular matrix expression on melt electrospun written scaffolds," *Acta Biomaterialia*, vol. 52, pp. 145–158, 2017. DOI: 10.1016/j.actbio.2016.12.040. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2016.12.040.

[319] J. Ren *et al.*, "Melt-electrospun polycaprolactone strontium-substituted bioactive glass scaffolds for bone regeneration," *Journal of Biomedical Materials Research - Part A*, vol. 102, no. 9, pp. 3140–3153, 2014. DOI: 10.1002/jbm.a.34985.

[320] J. Li, X. Liu, B. Zuo, and L. Zhang, "The Role of Bone Marrow Microenvironment in Governing the Balance between Osteoblastogenesis and Adipogenesis," *Aging and Disease*, vol. 7, no. 4, p. 514, 2016. DOI: 10.14336/AD.2015.1206. [Online]. Available: http://www.aginganddisease.org/EN/10.14336/AD.2015.1206.

[321] T. D. Brown, P. D. Dalton, and D. W. Hutmacher, "Direct writing by way of melt electrospinning," *Advanced Materials*, vol. 23, no. 47, pp. 5651–5657, 2011. DOI: 10.1002/adma.201103482.

[322] J. He, P. Xia, and D. Li, "Development of melt electrohydrodynamic 3D printing for complex microscale poly (ϵ-caprolactone) scaffolds.," *Biofabrication*, vol. 8, no. 3, p. 035 008, 2016. DOI: 10.1088/1758-5090/8/3/035008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/27490377.

[323] J. Lee, S. Y. Lee, J. Jang, Y. H. Jeong, and D. W. Cho, "Fabrication of patterned nanofibrous mats using direct-write electrospinning," *Langmuir*, vol. 28, no. 18, pp. 7267–7275, 2012. DOI: 10.1021/la3009249.

[324] S. Soliman *et al.*, "Multiscale three-dimensional scaffolds for soft tissue engineering via multimodal electrospinning," *Acta Biomaterialia*, vol. 6, no. 4, pp. 1227–1237, 2010. DOI: 10.1016/j.actbio.2009.10.051.

[325] F. Luongo, F. G. Mangano, A. Macchi, G. Luongo, and C. Mangano, "Custom-made synthetic scaffolds for bone reconstruction: A retrospective, multicenter clinical study on 15 patients.," *BioMed Research International*, vol. 2016, pp. 1–12, 2016. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=rzh%5C&AN=120216090%5C&site=eds-live%5C&scope=site.

[326] W.-F. Yang *et al.*, "Three-dimensional printing of patient-specific surgical plates in head and neck reconstruction: A prospective pilot study.," *Oral oncology*, vol. 78, no. January, pp. 31–36, 2018. DOI: 10.1016/j.oraloncology.2018.01.005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/29496055.

[327] C. Vaquette and J. Cooper-White, "A simple method for fabricating 3-D multilayered composite scaffolds," *Acta Biomaterialia*, vol. 9, no. 1, pp. 4599–4608, 2013. DOI: 10.1016/j.actbio.2012.08.015.

[328] S. A. Poursamar, J. Hatami, A. N. Lehner, C. L. Da Silva, F. C. Ferreira, and A. P. M. Antunes, "Gelatin porous scaffolds fabricated using a modified gas foaming technique: Characterisation and cytotoxicity assessment," *Materials Science and Engineering C*, vol. 48, pp. 63–70, 2015. DOI: 10.1016/j.msec.2014.10.074. [Online]. Available: http://dx.doi.org/10.1016/j.msec.2014.10.074.

[329] D. Bushan *et al.*, "Gelatin- hydroxyapatite- calcium sulphate based biomaterial for long term sustained delivery of bone morphogenic protein-2 and zoledronic acid for increased bone formation : In-vitro and in-vivo carrier properties," *Journal of Controlled Release*, vol. 272, no. December 2017, pp. 83–96, 2018. DOI: 10.1016/j.jconrel.2018.01.006. [Online]. Available: https://doi.org/10.1016/j.jconrel.2018.01.006.

[330] S. C. Ligon, R. Liska, J. Stampfl, M. Gurr, and R. Mülhaupt, "Polymers for 3D Printing and Customized Additive Manufacturing," *Chemical Reviews*, vol. 117, no. 15, pp. 10 212–10 290, 2017. DOI: 10.1021/acs.chemrev.7b00074. arXiv: NIHMS150003.

[331] F. Calignano *et al.*, "Overview on additive manufacturing technologies," *Proceedings of the IEEE*, vol. 105, no. 4, pp. 593–612, 2017. DOI: 10.1109/JPROC.2016.2625098.

[332] M. Aghazadeh, M. Samiei, E. Alizadeh, P. Porkar, M. Bakhtiyari, and R. Salehi, "Towards osteogenic bioengineering of dental pulp stem induced by sodium fluoride on hydroxyapatite based biodegradable polymeric scaffold," *Fibers and Polymers*, vol. 18, no. 8, pp. 1468–1477, 2017. DOI: 10.1007/s12221-017-7120-0.

[333] M. Á. Brennan *et al.*, "3d cell culture and osteogenic differentiation of human bone marrow stromal cells plated onto jet-sprayed or electrospun micro-fiber scaffolds," *Biomedical Materials*, vol. 10, no. 4, p. 045 019, Aug. 2015. DOI: 10.1088/1748-6041/10/4/045019. [Online]. Available: https://doi.org/10.1088/1748-6041/10/4/045019.

[334] T. G. Kim, H. J. Chung, and T. G. Park, "Macroporous and nanofibrous hyaluronic acid/collagen hybrid scaffold fabricated by concurrent electrospinning and deposition/leaching of salt particles," *Acta Biomaterialia*, vol. 4, no. 6, pp. 1611–1619, 2008. DOI: 10.1016/j.actbio.2008.06.008.

[335] F. A. Paskiabi, E. Mirzaei, A. Amani, M. A. Shokrgozar, R. Saber, and R. Faridi-Majidi, "Optimizing parameters on alignment of PCL/PGA nanofibrous scaffold: An artificial neural networks approach," *International Journal of Biological Macromolecules*, vol. 81, pp. 1089–1097, 2015. DOI: 10.1016/j.ijbiomac.2014.10.040. [Online]. Available: http://dx.doi.org/10.1016/j.ijbiomac.2014.10.040.

[336] A. Abarrategi *et al.*, "Biological properties of solid free form designed ceramic scaffolds with bmp-2: In vitro and in vivo evaluation," *PLoS ONE*, vol. 7, no. 3, 2012. DOI: 10.1371/journal.pone.0034117.

[337] A. Youssef, S. J. Hollister, and P. D. Dalton, "Additive manufacturing of polymer melts for implantable medical devices and scaffolds," *Biofabrication*, vol. 09, p. 012 002, 2017. DOI: 10.1088/1758-5090/aa5766.

[338] F. E. Udwadia and A. D. Schutte, "An Alternative Derivation of the Quaternion Equations of Motion for Rigid-Body Rotational Dynamics," *Journal of Applied Mechanics*, vol. 77, no. 4, Apr. 2010, 044505. DOI: 10.1115/1.4000917. eprint: https://asmedigitalcollection.asme.org/appliedmechanics/article-pdf/77/4/044505/5481265/044505_1.pdf. [Online]. Available: https://doi.org/10.1115/1.4000917.

[339] A. W. Pila, *Introduction To Lagrangian Dynamics*. Springer International Publishing, 2020, ISBN: 9783030223779. DOI: 10.1007/978-3-030-22378-6.

[340] S. Vlase, I. Negrean, M. Marin, and S. Năstac, "Kane's method-based simulation and modeling robots with elastic elements, using finite element method," *Mathematics*, vol. 8, no. 5, pp. 1–21, 2020. DOI: 10.3390/MATH8050805.

[341] Z. Yangwei and W. Liangming, "A Method to Establish the Dynamic Models of Multibody System Based on Kane's Equations," in *Proceedings of the 2nd International Conference on Modelling, Identification and Control*, Atlantis Press, 2015, pp. 21–26. DOI: 10.2991/mic-15.2015.5. [Online]. Available: https://doi.org/10.2991/mic-15.2015.5.

[342] Z. Hussain and N. Z. Azlan, "KANE's method for dynamic modeling," *Proceedings - 2016 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2016*, pp. 174–179, 2017. DOI: 10.1109/I2CACIS.2016.7885310.

[343] A. H. Bajodah and D. H. Hodges, "Canonical Kane's equations of motion for discrete dynamical systems," *AIAA Journal*, vol. 57, no. 10, pp. 4226–4240, 2019. DOI: 10.2514/1.J057603.

[344] N. Castro-González, M. F. Martínez-Serrano, and J. Robles, "Expressions for the Moore-Penrose inverse of block matrices involving the Schur complement," *Linear Algebra and Its Applications*, vol. 471, pp. 353–368, 2015. DOI: 10.1016/j.laa.2015.01.003. [Online]. Available: http://dx.doi.org/10.1016/j.laa.2015.01.003.

[345] M. B. Tasić, P. S. Stanimirović, and M. D. Petković, "Symbolic computation of weighted Moore-Penrose inverse using partitioning method," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 615–640, 2007. DOI: 10.1016/j.amc.2006.11.114. arXiv: 1104.1696.

[346] G. Taylor, "Disintegration of water drops in an electric field," *Royal Society*, vol. 280, no. 1382, 1964. DOI: 10.1098/rspa.1964.0151. [Online]. Available: https://doi.org/10.1098/rspa.1966.0227.

[347] K. Kim, H. Shim, and J. Kim, "Fiber Formation Model for Electrospinning . II . Stable Jet Voltage," *Fibers and Polymers*, vol. 17, no. 10, pp. 1634–1640, 2016. DOI: 10.1007/s12221-016-6035-5.

[348] J. H. He, Y. Wu, and W. W. Zuo, "Critical length of straight jet in electrospinning," *Polymer*, vol. 46, no. 26, pp. 12 637–12 640, 2005. DOI: 10.1016/j.polymer.2005.10.130.

[349] C. Ru, J. Chen, Z. Shao, M. Pang, and J. Luo, "A novel mathematical model for controllable near-field electrospinning," *AIP Advances*, vol. 4, no. 1, 2014. DOI: 10.1063/1.4861705.

[350] J. Ko, N. K. Mohtaram, P. C. Lee, S. M. Willerth, and M. B. Jun, "Mathematical model for predicting topographical properties of poly (ϵ-caprolactone) melt electrospun scaffolds including the effects of temperature and linear transitional speed," *Journal of Micromechanics and Microengineering*, vol. 25, no. 4, 2015. DOI: 10.1088/0960-1317/25/4/045018.

[351] T. D. Brown, P. D. Dalton, and D. W. Hutmacher, "Melt electrospinning today: An opportune time for an emerging polymer process," *Progress in Polymer Science*, vol. 56, pp. 116–166, 2016. DOI: 10.1016/j.progpolymsci.2016.01.001. [Online]. Available: http://dx.doi.org/10.1016/j.progpolymsci.2016.01.001.

[352] G. Zheng, W. Li, X. Wang, H. Wang, D. Sun, and L. Lin, "Experiment and Simulation of Coiled Nanofiber Deposition Behavior from Near-Field Electrospinning," in *Conference on Nano/Micro Engineered and Molecular Systems*, IEEE, 2010, pp. 284–288, ISBN: 9781424465453.

[353] T. A. Kowalewski, S. BŁoński, and S. Barral, "Experiments and modelling of electrospinning process," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 53, no. 4, pp. 385–394, 2005.

[354] O. Karatay and M. Doğan, "Design and Implementation of an Electrospinning System," in *Engineering of Nanobiomaterials: Applications of Nanobiomaterials*, Elsevier Inc., 2016, pp. 359–396, ISBN: 9780323417341. DOI: 10.1016/B978-0-323-41532-3.00012-9. [Online]. Available: https://doi.org/10.1016/B978-0-323-41532-3.00012-9.

[355] L. Xu, "Chaos , Solitons and Fractals A mathematical model for electrospinning process under coupled field forces," *Chaos, Solitons and Fractals*, vol. 42, no. 3, pp. 1463–1465, 2009. DOI: 10.1016/j.chaos.2009.03.054. [Online]. Available: http://dx.doi.org/10.1016/j.chaos.2009.03.054.

[356] N. Ismail, F. J. Maksoud, N. Ghaddar, K. Ghali, and A. Tehrani-Bagha, "Simplified modeling of the electrospinning process from the stable jet region to the unstable region for predicting the final nanofiber diameter," *Journal of Applied Polymer Science*, vol. 133, no. 43, pp. 1–11, 2016. DOI: 10.1002/app.44112.

[357] S. Rafiei, S. Maghsoodloo, B. Noroozi, V. Mottaghitalab, and A. K. Haghi, "Mathematical modeling in electrospinning process of nanofibers: A detailed review," *Cellulose Chemistry and Technology*, vol. 47, no. 5-6, pp. 323–338, 2013.

[358] Q. Hu, Y. Jia, and S. Xu, "An improved Kane's method for multibody dynamics," *Lixue Xuebao/Chinese Journal of Theoretical and Applied Mechanics*, vol. 43, no. 5, pp. 968–972, 2011.

[359] F. E. Udwadia, "Equations of motion for constrained multibody systems and their control," *Journal of Optimization Theory and Applications*, vol. 127, no. 3, pp. 627–638, 2005. DOI: 10.1007/s10957-005-7507-8.

[360] F. E. Udwadia and A. D. Schutte, "Equations of motion for general constrained systems in lagrangian mechanics," *Acta Mechanica*, vol. 213, no. 1-2, pp. 111–129, 2010. DOI: 10.1007/s00707-009-0272-2.

[361] F. E. Udwadia and T. Wanichanon, "On general nonlinear constrained mechanical systems," *Numerical Algebra, Control and Optimization*, vol. 3, no. 3, pp. 425–443, 2013. DOI: 10.3934/naco.2013.3.425.

[362] F. E. Udwadia, "A new approach to stable optimal control of complex nonlinear dynamical systems," *Journal of Applied Mechanics, Transactions ASME*, vol. 81, no. 3, 2014. DOI: 10.1115/1.4024874.

[363] C. P. Carroll and Y. L. Joo, "Discretized modeling of electrically driven viscoelastic jets in the initial stage of electrospinning," *Journal of Applied Physics*, vol. 094315, no. March, 2011. DOI: 10.1063/1.3582119.

[364] E. G. Hemingway and O. M. O'Reilly, "Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments," *Multibody System Dynamics*, vol. 44, no. 1, pp. 31–56, 2018. DOI: 10.1007/s11044-018-9620-0. [Online]. Available: http://dx.doi.org/10.1007/s11044-018-9620-0.

[365] J. Xu and K. H. Halse, "Dual Quaternion Variational Integrator for Rigid Body Dynamic Simulation," —, 2016. arXiv: 1611.00616. [Online]. Available: http://arxiv.org/abs/1611.00616.

[366] N. D. Zoric, M. P. Lazarevic, and A. M. Simonovic, "Multi-body kinematics and dynamics in terms of quaternions: Langrange formulation in covariant form - Rodriguez approach," *FME Transactions*, vol. 38, no. 1, pp. 19–28, 2010.

[367] T. CZYZEWSKI, "Quaternions – Alternative for Euler Angles in Kinematic Analysis of Machine Mechanisms," *Technical Transactions Mechanics*, 2015.

[368] M. Möller and C. Glocker, "Rigid body dynamics with a scalable body, quaternions and perfect constraints," *Multibody System Dynamics*, vol. 27, no. 4, pp. 437–454, 2012. DOI: 10.1007/s11044-011-9276-5.

[369] R. Campa and K. Camarillo, "Unit Quaternions: A Mathematical Tool for Modeling, Path Planning and Control of Robot Manipulators," in *Robot Manipulators*, InTech, 2008, pp. 21–48. DOI: 10.5772/6197.

[370] B. Graf, "Quaternions and dynamics," —, 2008. arXiv: 0811.2889. [Online]. Available: http://arxiv.org/abs/0811.2889.

[371] M. Arribas, A. Elipe, and M. Latasa, "Quaternions and the rotation of a rigid body," *Celestial Mechanics and Dynamical Astronomy*, vol. 96, pp. 239–251, Nov. 2006. DOI: 10.1007/s10569-006-9037-6.

[372] F. M. Wunner, J. Maartens, O. Bas, K. Gottschalk, E. M. De-juan-pardo, and D. W. Hutmacher, "Electrospinning writing with molten poly (e -caprolactone) from different directions – Examining the effects of gravity," *Materials Letters*, vol. 216, pp. 114–118, 2018. DOI: 10.1016/j.matlet.2017.12.079. [Online]. Available: https://doi.org/10.1016/j.matlet.2017.12.079.

[373] M. Šimko, J. Erhart, and D. Lukáš, "A mathematical model of external electrostatic field of a special collector for electrospinning of nanofibers," *Journal of Electrostatics*, vol. 72, no. 2, pp. 161–165, 2014. DOI: 10.1016/j.elstat.2014.01.003.

[374] D. Gutierrez-Lemini, "Engineering viscoelasticity," *Engineering Viscoelasticity*, pp. 1–353, 2014. DOI: 10.1007/978-1-4614-8139-3.

[375] H. F. Brinson and L. C. Brinson, *Polymer engineering science and viscoelasticity: An introduction, Second edition*. Springer US, 2015, pp. 1–482, ISBN: 9781489974853. DOI: 10.1007/978-1-4899-7485-3.

[376] A. Pipkin, *Lectures on Viscoelasticity Theory*, 2nd ed. New York: Springer-Verlag New York, 1986, p. 188, ISBN: 9781626239777.

[377] Q. Hao and S. Greenhalgh, "The generalized standard-linear-solid model and the corresponding viscoacoustic wave equations revisited," *Geophysical Journal International*, vol. 219, no. 3, pp. 1939–1947, 2019. DOI: 10.1093/gji/ggz407.

[378] D. H. Reneker and A. L. Yarin, "Electrospinning jets and polymer nanofibers," vol. 49, pp. 2387–2425, 2008. DOI: 10.1016/j.polymer.2008.02.002.

[379] P. Mann, *Lagrangian and Hamiltonian dynamics*. Oxford University Press, 2018, pp. 1–538, ISBN: 9780198822370. DOI: 10.1093/oso/9780198822370.001.0001. [Online]. Available: https://doi.org/10.1093/oso/9780198822370.001.0001.

[380] P. Flores, M. MacHado, E. Seabra, and M. Tavares Da Silva, "A parametric study on the baumgarte stabilization method for forward dynamics of constrained multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 6, no. 1, 2011. DOI: 10.1115/1.4002338.

[381] M. Hajzman and P. Polach, "Application of stabilization techniques in the dynamic analysis of multibody systems," *Applied and Computational Mechanics*, vol. 1, no. November 2007, pp. 479–488, 2007.

[382] L. Guizhi and L. Rong, "Determination of Stability Correction Parameters for Dynamic Equations of Constrained Multibody Systems," *Mathematical Problems in Engineering*, vol. 2018, 2018. DOI: 10.1155/2018/8945301.

[383] F. M. Amirouche and C. W. Tung, "Regularization and stability of the constraints in the dynamics of multibody systems," *Nonlinear Dynamics*, vol. 1, no. 6, pp. 459–475, 1990. DOI: 10.1007/BF01856949.

[384] I. Liashenko, A. Hrynevich, and P. D. Dalton, "Designing Outside the Box : Unlocking the Geometric Freedom of Melt Electrowriting using Microscale Layer Shifting," DOI: 10.1002/adma.202001874.

[385] H. Xu, M. Yamamoto, and H. Yamane, "Melt electrospinning: Electrodynamics and spinnability," *Polymer*, vol. 132, pp. 206–215, 2017. DOI: 10.1016/j.polymer.2017.11.006. [Online]. Available: https://doi.org/10.1016/j.polymer.2017.11.006.

[386] Y.-X. Wan, Y.-P. Xiong, and S.-M. Zhang, "Temperature Effect on Dynamic Properties of Magnetorheological Elastomers," in *3rd Annual International Conference on Advanced Material Engineering*, vol. 110, Atlantis Press., 2017, pp. 415–420. DOI: 10.2991/ame-17.2017.68.

[387] M. E. Yildiz and J. L. Kokini, "Determination of Williams–Landel–Ferry constants for a food polymer system: Effect of water activity and moisture content," *Journal of Rheology*, vol. 45, no. 4, pp. 903–912, 2001. DOI: 10.1122/1.1380425.

[388] A. Álvarez-Vázquez, A. Fernández-Canteli, E. Castillo, F. Pelayo, M. Muñiz-Calvente, and M. J. Lamela, "A time- and temperature-dependent viscoelastic model based on the statistical compatibility condition," *Materials and Design*, vol. 193, p. 108 828, 2020. DOI: 10.1016/j.matdes.2020.108828. [Online]. Available: https://doi.org/10.1016/j.matdes.2020.108828.

[389] A. D. Drozdov, "Effect of temperature on the viscoelastic and viscoplastic behavior of polypropylene," *Mechanics of Time-Dependent Materials*, vol. 14, no. 4, pp. 411–434, 2010. DOI: 10.1007/s11043-010-9118-5.

[390] L. E. Freed *et al.*, "Biodegradable polymer scaffold for tissue engineering," *Nature Biotechnology*, vol. 12, no. 1, pp. 1119–1124, 1994. DOI: 10.1038/ng1294-340.

[391] S. Bose, M. Roy, and A. Bandyopadhyay, "Recent advances in bone tissue engineering scaffolds," *Trends in Biotechnology*, vol. 30, no. 10, pp. 546–554, 2012. DOI: 10.1016/j.tibtech.2012.07.005. arXiv: NIHMS150003. [Online]. Available: http://dx.doi.org/10.1016/j.tibtech.2012.07.005.

[392] S. Sahoo, L. T. Ang, J. C.-H. Goh, and S.-L. Toh, "Growth factor delivery through electrospun nanofibers in scaffolds for tissue engineering applications," *Journal of Biomedical Materials Research Part A*, vol. 93A, no. 4, pp. 1539–1550, 2010. DOI: https://doi.org/10.1002/jbm.a.32645. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jbm.a.32645. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jbm.a.32645.

[393] A. Wubneh, C. Ayranci, and C.-I. Kim, "A novel theoretical model development and simulation of melt-electrospinning using kane's and udwadia-kalaba methods.," *ADVANCED THEORY AND SIMULATIONS*, p. 2 100 278, 2022. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true%5C&db=edswsc%5C&AN=000746212600001%5C&site=eds-live%5C&scope=site.

[394] C. M. Roithmayr and D. H. Hodges, *Dynamics*. Cambridge University Press, Mar. 2015. DOI: 10.1017/cbo9781139047524. [Online]. Available: https://doi.org/10.1017/cbo9781139047524.

[395] A. Wubneh, C. K. Au, and Y.-S. Ma, "Feature transformation from configuration of open-loop mechanisms into linkages with a case study," in *Springer Series in Advanced Manufacturing*, London: Springer London, 2013, pp. 275–302.

[396] A. Wubneh and Y.-S. Ma, "Feature-Based mechanism design," in *Springer Series in Advanced Manufacturing*, London: Springer London, 2013, pp. 303–352.

[397] D. C. Montgomery, *Design and analysis of experiments*. John Wiley & Sons, Inc., 2013, ISBN: 9781118146927. [Online]. Available: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat03710a&AN=alb.5871247&site=eds-live&scope=site.

[398] M. F. P. Costa and C. Ribeiro, "Generalized fractional Maxwell model: Parameter estimation of a viscoelastic material," *AIP Conference Proceedings*, vol. 1479, no. 1, pp. 790–793, 2012. DOI: 10.1063/1.4756256.

[399] A. Bonfanti, J. L. Kaplan, G. Charras, and A. Kabla, "Fractional viscoelastic models for power-law materials," *Soft Matter*, vol. 16, no. 26, pp. 6002–6020, 2020. DOI: 10.1039/d0sm00354a. arXiv: 2003.07834.

[400] S. Katsourinis and E. Kontou, "Fractional viscoelastic models for interconverting linear viscoelastic functions of various polymeric structures," *Rheologica Acta*, vol. 58, no. 5, pp. 307–320, 2019. DOI: 10.1007/s00397-019-01146-y.

[401] C. Li, F. Mainardi, and G. Spada, "Special issue: Perspectives on Fractional Dynamics and Control Creep, Relaxation and Viscosity Properties for Basic Fractional Models in Rheology," *The European Physical Journal, Special Topics*, vol. 193, pp. 133–160, 2011. arXiv: 1110.3400v1. [Online]. Available: http://www.fracalmo.org.

[402] F. Chen, G. Hochleitner, T. Woodfield, J. Groll, P. D. Dalton, and B. G. Amsden, "Additive manufacturing of a photo-cross-linkable polymer via direct melt electrospinning writing for producing high strength structures," *Biomacromolecules*, vol. 17, no. 1, pp. 208–214, Dec. 2015. DOI: 10.1021/acs.biomac.5b01316. [Online]. Available: https://doi.org/10.1021/acs.biomac.5b01316.

[403] Y. Jin *et al.*, "Fabrication of heterogeneous scaffolds using melt electrospinning writing: Design and optimization," *Materials and Design*, vol. 185, p. 108 274, Jan. 2020. DOI: 10.1016/j.matdes.2019.108274. [Online]. Available: https://doi.org/10.1016/j.matdes.2019.108274.

[404] S. Loewner *et al.*, "Recent advances in melt electro writing for tissue engineering for 3d printing of microporous scaffolds for tissue engineering," *Frontiers in Bioengineering and Biotechnology*, vol. 10, Aug. 2022. DOI: 10.3389/fbioe.2022.896719. [Online]. Available: https://doi.org/10.3389/fbioe.2022.896719.

[405] Y.-S. Park *et al.*, "Near-field electrospinning for three-dimensional stacked nanoarchitectures with high aspect ratios," *Nano Letters*, vol. 20, no. 1, pp. 441–448, Nov. 2019. DOI: 10.1021/acs.nanolett.9b04162. [Online]. Available: https://doi.org/10.1021/acs.nanolett.9b04162.

[406] W. King and G. Bowlin, "Near-field electrospinning and melt electrowriting of biomedical polymers—progress and limitations," *Polymers*, vol. 13, no. 7, p. 1097, Mar. 2021. DOI: 10.3390/polym13071097. [Online]. Available: https://doi.org/10.3390/polym13071097.

[407] Z. G. Davis, A. F. Hussain, and M. B. Fisher, "Processing variables of direct-write, near-field electrospinning impact size and morphology of gelatin fibers," *Biomedical Materials*, vol. 16, no. 4, p. 045 017, May 2021. DOI: 10.1088/1748-605x/abf88b. [Online]. Available: https://doi.org/10.1088/1748-605x/abf88b.

[408] X. Mu, Y. Zheng, X. Li, B. Xin, and L. Lin, "Effects of temperature on melt electrospinning: Experiment and simulation study," *Fibers and Polymers*, vol. 22, no. 4, pp. 964–971, Mar. 2021. DOI: 10.1007/s12221-021-0465-4. [Online]. Available: https://doi.org/10.1007/s12221-021-0465-4.

[409] A. Bachs-Herrera, O. Yousefzade, L. J. del Valle, and J. Puiggali, "Melt electrospinning of polymers: Blends, nanocomposites, additives and applications," *Applied Sciences*, vol. 11, no. 4, p. 1808, Feb. 2021. DOI: 10.3390/app11041808. [Online]. Available: https://doi.org/10.3390/app11041808.

[410] Y. Liu, R. Deng, M. Hao, H. Yan, and W. Yang, "Orthogonal design study on factors effecting on fibers diameter of melt electrospinning," *Polymer Engineering and Science*, vol. 50, no. 10, pp. 2074–2078, Sep. 2010. DOI: 10.1002/pen.21753. [Online]. Available: https://doi.org/10.1002/pen.21753.

Appendix A: MATLAB® Codes

A.1 Scripts

Listing A.1: dashboard

```
%DASHBOARD is a Matlab script designed to temporarily serve the role of a GUI until
% a proper GUI is developed for this application. This script provides fields for
% solution definition, equipment parameter setting, and options to select which
% forces to include in the simulation.
%
% This is also where the file name and director for g-codes are declared. The
% ultimate goal is to wrap this script into a GUI.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%
%================================================================================
clear all
close all
clc
tic;
%% General simulation session information holder
sessionInfo.name = 'Cleaning';
sessionInfo.allSessionsPath = 'C:\Users\Workstation\Desktop\Simulation_Results';
% This is where all other similar sessions are saved
sessionInfo.thisSessionPath = [sessionInfo.allSessionsPath '\' sessionInfo.name];
% This is the complete path to the session folder itself
sessionInfo.simulationDate = ...
   datetime('now','TimeZone','local','Format','d-MMM-y HH:mm:ss Z');
sessionInfo.runEquipmentSettingFromFile = yes;
sessionInfo.runEntireTable = no;
sessionInfo.testRunNumber = 10;
sessionInfo.isNozzleMoving = yes;
sessionInfo.gCodeFilePath = ...
   'C:\Users\Workstation\Documents\Inkscape\MFCL_0003.ngc';
% navigate to the location, get a copy and save it in the solution directory
sessionInfo.DOEdataFilePath = [pwd '\Experiment_data.xlsx'];
% temporarily saved in the current Matlab work folder
sessionInfo.inpectSegmentResults = no;
%% G-code Settings
gCode.feedRateOveride = 500; % (mm/min)
if gCode.feedRateOveride > 10000
```

```
   % If the feed rate is faster than 10000 mm/min, make the rapid movement speed
   % same as the feed rate.
   gCode.rapidOveride = gCode.feedRateOveride;
else
   gCode.rapidOveride = 10000; %
end
gCode.filePath = sessionInfo.gCodeFilePath;
gCode.totalTime = 0;
gCode.gCodeKinematicsData = [];
gCode.nozTPVA = [];
gCode.gCodeTable = [];
gCode.gCode_struct.unit = mm;
if lower(sessionInfo.isNozzleMoving) == yes
   gCode = gCodeKinematicsFcn(gCode);
   fprintf(['Total g-code running time is ' num2str(gCode.totalTime) ' s.'])
end
sessionInfo.simulationTimeLength = 0.15;   %gCode.totalTime;
sessionInfo.simulationSegmentTimeLimit = 0.1;
%% Equipment Setting
equipment_setting.platformSize = [160 140]; % (mm x mm)
equipment_setting.flowrate = 1;             % (ml/hr)
equipment_setting.voltage = 15;             % (kV)
equipment_setting.collectordistance = 50;   % (mm)
equipment_setting.meltTemp = 240;           % deg C)
equipment_setting.ambTemp = 60;             % (deg. Celcius)
equipment_setting.directionOfSpinning = 0;  % (deg) 0 for downward nozzle
equipment_setting.nozzleDiameter = 0.2;     % (mm)
%% Solver Setting
solver.numOfStepsPerEjection = 10;
solver.Baumgarte.alpha_baumgarte = 10000;
solver.Baumgarte.beta_baumgarte = 10000;
solver.stuckBeadLimit = 1;
%% Forces to include (... 1 to include, 0 to exclude)
forces.includeGravitationalForces = 1;
forces.includeViscoelasticLinear = 1;
forces.includeViscoelasticTorsional = 1;
forces.includeCoulombForces = 1;
forces.includeElectricFieldForces = 1;
%% Material and process parameters
load('elasticParametersFile.mat');
material.scaleKt = 1;
material.scaleCt = 1;
material.ambientTempFactor.E0 = 1;
material.ambientTempFactor.E1 = 1;
material.ambientTempFactor.eta = 1;
%% Experiment Validation Options
%{
If this option is selected, the program reads the equipment setting values from a
file. Further options could be selected to run a single row from the file data or
the entire table, with the latter results saved back to a file specified at the
end of the simulation.
%}
if ~isfolder(sessionInfo.allSessionsPath)
   % .. if the high-level folder, where all other similar sessions are saved,
```

```
92      % isn't already existing, create one.
93      mkdir(sessionInfo.allSessionsPath);
94  end
95  if isfolder(sessionInfo.thisSessionPath)
96      % if the session's folder already exists, delete it and its contents
97      rmdir (sessionInfo.thisSessionPath, s);
98  end
99  mkdir(sessionInfo.thisSessionPath);
100 if lower(sessionInfo.runEquipmentSettingFromFile) == yes
101     dataMatrix = readmatrix(sessionInfo.DOEdataFilePath);
102     [numberOfRuns,lastCol] = size(dataMatrix);
103     if lower(sessionInfo.runEntireTable) == no
104         startRow = sessionInfo.testRunNumber;
105         endRow = sessionInfo.testRunNumber;
106     elseif lower(sessionInfo.runEntireTable) == yes
107         startRow = 1;
108         endRow = numberOfRuns;
109     else
110         error(Unknown experimental data range option entered!  )
111     end
112     for DOErunNumber = startRow:1:endRow
113         solutionPath = [sessionInfo.thisSessionPath '\Run' num2str(DOErunNumber)];
114         solver.DOErunNumber = DOErunNumber;
115         equipment_setting.flowrate = dataMatrix(DOErunNumber,3);
116         equipment_setting.voltage = dataMatrix(DOErunNumber,4);
117         equipment_setting.collectordistance = dataMatrix(DOErunNumber,5);
118         equipment_setting.meltTemp = dataMatrix(DOErunNumber,6);
119         switch equipment_setting.meltTemp
120             case 200
121                 elasticParameters = elasticParametersStruct.temp200C;
122             case 215
123                 elasticParameters = elasticParametersStruct.temp215C;
124             case 230
125                 elasticParameters = elasticParametersStruct.temp230C;
126             case 240
127                 elasticParameters = elasticParametersStruct.temp240C;
128             case 250
129                 elasticParameters = elasticParametersStruct.temp250C;
130         end
131         material.E0t = elasticParameters(1,1);
132         material.E1t = elasticParameters(2,1);
133         material.etat = elasticParameters(2,2);
134         if isfolder(solutionPath)
135             rmdir (solutionPath, s);
136         end
137         mkdir(solutionPath);
138         solver.solutionPath = solutionPath;
139         param = ParametersFcn(equipment_setting, ...
140             solver, ...
141             forces, ...
142             material);
143         param.sessionInfo = sessionInfo;
144         param.gCode = gCode;
145         param.equipment_setting = equipment_setting;
```

```
146         % for better readability of the setting values
147         if lower(sessionInfo.isNozzleMoving) == yes
148             figure_gCode = gCodeplotForInspectionFcn(param);
149             pause(1);
150             savefig([solutionPath '/Gcode_Inspection.fig']);
151             copyfile(sessionInfo.gCodeFilePath, solutionPath)
152         end
153         [simulationSummary, updatedParam] = solverFcn(param, DOErunNumber, endRow);
154         simSummary = simulationSummary;
155         param = updatedParam;
156         dataMatrix(DOErunNumber,lastCol+1:lastCol+2) = ...
157             [1000*simSummary.fiberChar.maximumCollectionDiameter, ...
158             1000000*simSummary.fiberChar.averageFiberDiameter];
159     end
160     if lower(sessionInfo.runEntireTable) == yes
161         observedCollectionD = dataMatrix(:,7);
162         observedFiberD = dataMatrix(:,8);
163         predictedCollectionD = dataMatrix(:,lastCol+1);
164         predictedFiberD = dataMatrix(:,lastCol+2);
165         errorCollectionD = observedCollectionD - predictedCollectionD;
166         errorFiberD = observedFiberD - predictedFiberD;
167         pctErrorCollectionD = abs(errorCollectionD ./ observedCollectionD);
168         pctErrorFiberD = abs(errorFiberD ./ observedFiberD);
169         dataMatrix(:,lastCol+3:lastCol+4) = [pctErrorCollectionD,pctErrorFiberD];
170         comparisionTable = array2table(dataMatrix, 'VariableNames', ...
171             { 'StdOrder',...
172             'RunOrder',...
173             'Flowrate',...
174             'Voltage',...
175             'Distance',...
176             'MeltTemp',...
177             'Observed_Collection_Diameter',...
178             'Observed_Fiber_Diameter',...
179             'Predicted_Collection_Diameter',...
180             'Predicted_Fiber_Diameter',...
181             'Percentage_Error_Collection_Diameter',...
182             'Percentage_Error_Fiber_Diameter'
183             });
184         writetable(comparisionTable, 'ComparisionTable_REV01.xlsx')
185     end
186     if lower(sessionInfo.runEntireTable) == no
187         elapsedTime = toc;
188         Duration = duration(seconds(elapsedTime));
189         Duration.Format = 'hh:mm:ss.S';
190         disp(' ==================== SIMULATION STATUS ========================')
191         disp(['  Based on DOE Run# ' num2str(DOErunNumber)]);
192         disp(sessionInfo.simulationDate);
193         disp(' ------------------------------------------------------------')
194         disp(' ')
195         disp(['  Virtual current simulation time: ' ...
196             num2str(simSummary.param.sessionInfo.simulationTimeLength,'%.2f') ...
197             ' s. (completed)' ]);
198         disp(['  Current simulation segment:      ' ...
199             num2str(simSummary.param.solver.numOfSegments) ' of ' ...
```

```
        num2str(simSummary.param.solver.numOfSegments)]);
    disp(['  Number of beads ejected:         ' ...
        num2str(simSummary.param.solver.numOfSegments)]);
    percentage = 100;
    disp(['  Solver status:                   ' ...
        num2str(percentage,'%.0f') ' % completed']);
    disp(' ')
    disp(['  Simulation segment length:                  ' ...
        num2str(simSummary.param.sessionInfo.simulationSegmentTimeLimit,...
        '%.2f') ' s']);
    disp(['  Numerical time step:                        ' ...
        num2str(simSummary.param.solver.timeStep,'%.5f') ' s']);
    disp(['  Virtual time length the simulation run for:  ' ...
        num2str(simSummary.param.sessionInfo.simulationTimeLength) ' s'])
    disp(['  Elapsed time (actual time it took to solve): ' char(Duration)])
    disp(['  Feedrate:                                   ' ...
        num2str(simSummary.param.gCode.feedRateOveride)]);
    disp(' ')
    disp('————————————————————————————————————————————————————————————————————————————————')
    disp(' ')
    disp('PROCESS SETTING VALUES')
    disp(['      Flow rate:                          ' ...
        num2str(simSummary.param.equipment_setting.flowrate) ' ml/hr'])
    disp(['      Applied voltage:                    ' ...
        num2str(simSummary.param.equipment_setting.voltage) ' kV'])
    disp(['      Collector distance:                 ' ...
        num2str(simSummary.param.equipment_setting.collectordistance) ' mm'])
    disp(['      Melt temperature:                   ' ...
        num2str(simSummary.param.equipment_setting.meltTemp) ' deg. C'])
    disp(['      (Correspoinding experiment run #:) ' ...
        num2str(simSummary.param.sessionInfo.testRunNumber)])
    disp(' ')
    disp('————————————————————————————————————————————————————————————————————————————————')
    disp(' ')
    disp('ELASTIC PARAMETER VALUES')
    disp(['      E_0   ' num2str(simSummary.param.material.E0t) ' Pa'])
    disp(['      E_1   ' num2str(simSummary.param.material.E1t) ' Pa'])
    disp(['      n_1   ' num2str(simSummary.param.material.etat) ' Pa.s'])
    disp('————————————————————————————————————————————————————————————————————————————————')
    disp(' ')
    measuredCollectionDiameter = dataMatrix(DOErunNumber,7);
    measuredFiberDiameter = dataMatrix(DOErunNumber,8);
    Headers2 = {'MODEL PREDICTED','EXPERIMENTALLY MEASURED', 'DIFFERENCE'};
    rowNames2 = {'Collection diameter', 'Fiber diameter'};
  end
end
disp(' ========================= SIMULATION COMPLETED ========================')
%% Sound when solution completes
sound(sin(1:3000))
sound(sin(1:3000))
sound(sin(1:3000))
animateFcn(solutionPath,yes)
```

A.2 Classes

Listing A.2: Body class

```
classdef BodyClass < matlab.mixin.SetGet
   % BodyClass is a class designed to represent the beads in the system as
   % components of a multi-body system. While its member attributes store material,
   % inertial, and kinematic data about bead instances, its member functions
   % compute operations to determine the beads' kinematic states.
   %
   %  Author: Abiy Wubneh
   %  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
   %
   properties
      % IDENTIFIERS
      bodyType = 'Unnamed body type';
      % FIBER CHARACTERSTICS
      beadRadius = 0;
      mass = 0;
      Jo = 0;
      % MATERIAL PROPERTIES
      materialName = 'Unnamed material';
      density = 0;
      % KINEMATIC PARAMETERS
      % Generalized coordinates and their derivatives
      pos = [0 0 0]';
      quat = [1 0 0 0]';
      pos_dt = [0 0 0]';
      quat_dt = [0 0 0 0]';
   end
   methods
      %%
      function thisBody = BodyClass(param)
         thisBody.bodyType = param.bead.bodyType;
         thisBody.beadRadius = param.bead.radius;
         thisBody.mass = param.bead.mass;
         thisBody.Jo = param.bead.Jo;
         thisBody.materialName = param.material.materialName;
         thisBody.density = param.material.density;
      end
      %%
      function E = EMatFn(THISBODY)
         % Generates the E matrix of a given body from the quaternions
         q = THISBODY.quat;
         E = [-q(2) q(1) q(4) -q(3);
            -q(3) -q(4) q(1) q(2);
            -q(4) q(3) -q(2) q(1)];
      end
      %%
      function G = GMatFn(THISBODY)
         % Generates the G matrix from the quaternions
         q = THISBODY.quat;
         G = [-q(2) q(1) -q(4) q(3);
            -q(3) q(4) q(1) -q(2);
            -q(4) -q(3) q(2) q(1)];
```

```
52        end
53        %%
54        function E_dt = EdtMatFn(THISBODY)
55           % Generates the time derivative of the E matrix
56           q_dt = THISBODY.quat_dt;
57           E_dt = [-q_dt(2) q_dt(1) q_dt(4) -q_dt(3);
58              -q_dt(3) -q_dt(4) q_dt(1) q_dt(2);
59              -q_dt(4) q_dt(3) -q_dt(2) q_dt(1)];
60        end
61        %%
62        function G_dt = GdtMatFn(THISBODY)
63           % Generates the time derivative of the G matrix
64           q_dt = THISBODY.quat_dt;
65           G_dt = [-q_dt(2) q_dt(1) -q_dt(4) q_dt(3);
66              -q_dt(3) q_dt(4) q_dt(1) -q_dt(2);
67              -q_dt(4) -q_dt(3) q_dt(2) q_dt(1)];
68        end
69        %%
70        function R = RMatFn(THISBODY)
71           %{
72              RMatFn Rotation Matrix Function: Generates the rotation matrix R  of
73              the bead instance.
74
75              Use-case 1, Rotation:
76              This matrix can rotate a vector in the reference frame N to a new
77              orientation dictated by the quaternion angle.
78
79              The result is a rotated vector (another vector) expressed in the same
80              reference frame and using the same coordinates as the original vector.
81
82              E.g., given a position vector N_Poriginal_n, pointing from the origin
83              to point Poriginal in N, R can be used to rotate this vector to a new
84              point Protated. The resulting position vector N_Protated_n is given by:
85                              N_Protated_n = R*N_Poriginal_n
86
87              Use-case 2, Transformation
88              2. a) Inverse transformation
89              R can also express a given vector expressed in body-fixed coordinates
90              (b) in terms of inertial reference coordinate system (n). Note that the
91              reference frame doesn't change, just the coordinates!
92
93              E.g., Let a position vector ponting from the origin of the  body-fixed
94              reference frame B to a Point P be given by N_Poriginal_b (in the
95              inertial reference frame (N) but written in body-fixed coordinates
96              (b)). The expression of this position vector in the inertial reference
97              coordinates (n) is then given by:
98                              N_Poriginal_n = R*N_Poriginal_b
99
100             2. b) Forward transformation
101             If the original vector was expressed in the inertial coordinates (n),
102             and one was interested in expressing this vector in the body-fixed
103             coordinates (b), the transpose of R can be used in the calculation
104             instead is as follows:
105                             N_Poriginal_b = R'*N_Poriginal_n
```

```
106
107            Reference on this conversion formula can be found at:
108            https://www.mathworks.com/help/fusion/ref/quaternion.rotmat.html
109            https://www.andre-gaschler.com/rotationconverter/
110            https://www.euclideanspace.com/maths/geometry/rotations/
111            conversions/quaternionToMatrix/index.htm
112         %}
113         q = THISBODY.quat;
114         R = [2*(q(1)^2+q(2)^2)-1,2*(q(2)*q(3)-q(1)*q(4)),2*(q(2)*q(4)+q(1)*q(3));
115            2*(q(2)*q(3)+q(1)*q(4)),2*(q(1)^2+q(3)^2)-1,2*(q(3)*q(4)-q(1)*q(2));
116            2*(q(2)*q(4)-q(1)*q(3)),2*(q(3)*q(4)+q(1)*q(2)),2*(q(1)^2+q(4)^2)-1];
117         % Alternatively R can be writtent as: R = G*E.'
118      end
119      %%
120      function [Mp, Mq, Mi] = massMatrixFn(THISBODY)
121         %{
122            massMatrixFn: Mass Matrix Function:
123            Constructs the inertial matrix of a bead
124         %}
125         Mp = THISBODY.mass*eye(3);
126         Mq = 4*THISBODY.Jo*eye(4);
127         Mi = blkdiag(Mp,Mq);
128      end
129      %%
130      function [N_omegBody_n, N_omegBody_b] = omegFn(THISBODY)
131         %{
132            omegaFn: Omeag function calculates the body's angular velocity in rad/s
133            with respect to the inertial reference frame N. The components of this
134            velocity projected onto the inertial reference frame are given by:
135                              N_omegaBody_n
136            And the components of the angular velocity projected onto the body-
137            fixed reference frame are given by:
138                              N_omegaBody_b
139         %}
140         E = THISBODY.EMatFn;
141         G = THISBODY.GMatFn;
142         q_dt = THISBODY.quat_dt;
143         N_omegBody_b = 2*E*q_dt;
144         N_omegBody_n = 2*G*q_dt;            % Ref. [ 1 ]
145      end
146
147      %%
148      function [A,b_baumgarte] = AbFn(THISBODY,Alpha,Beta)
149         %{
150            AbFn: A matrix and b vector generator function:
151            Generates the A matrix and b vector of the quaternion constraint
152            equation.i.e.,
153                           A*q_dd = b
154            is the constraint equation due to the identity requirement set on
155            components of the quaternion. i.e.,
156                q = [q0; q1; q2; q3]
157                q0^2 + q1^2 + q2^2 + q3^2 = 1
158
159            Original holonomic constraint
```

```
                constraint: q0^2 + q1^2 + q2^2 + q3^2  - 1 = 0

            The time derivative of the holonomic constraint:
            constraint_dt = 2(q0q0_dt + q1q1_dt + q2q2_dt + q3q3_dt) = 0
                Or in matrix form:
                  constraint_dt = 2*q'q_dt = 0

                Second-time derivative of the holonomic constraint:
                  constrain_dtt = 2*qT*q_dd = -2*norm(q_d)^2    [ 1 ]

                Holonomic constrain_ddt: A*quat_ddt = b;
                where,  A = 2*qT  and  b = -2*norm(q_d)^2

                Stabilized constraint:
                constraint_ddt = A*quat_ddt - b_baumgarte
                where b_baumgarte = b - alpha1*holonomicCsontraintValue -
                alpha2*nonHolonomicConstraint_dtValue

            Original holonomic constraint in phi(q,t)  = 0 format (ideally, this
            expression should evaluate to zero, but normally is a non-zero value,
            representing the error/drift).
        %}
        q = THISBODY.quat;
        q_dt = THISBODY.quat_dt;
        holonomicConstraint = sum(q.*q) - 1;
        holonomicConstraint_dt = 2*transpose(q)*q_dt;
        A = 2*[0 0 0 q(1) q(2) q(3) q(4)];
        b = -2*(q_dt(1)^2 + q_dt(2)^2 + q_dt(3)^2 + q_dt(4)^2);
        b_baumgarte = b - 2*Alpha*holonomicConstraint_dt - ...
            Beta^2*holonomicConstraint;         % Ref. [ 2 ]
    end
  end
end
%{
REFERENCES:
[ 1 ]. Firdaus Udwadia, Aaron Schutte. An Alternative
Derivation of the Quaternion Equations of Motion for Rigid-Body
Rotational Dynamics. Journal of Applied Mechanics, American
Society of Mechanical Engineers, 2010, 77 (4), pp.044505.
ff10.1115/1.4000917ff. ffhal-01352566ff
%}
```

Listing A.3: Joint class

```
classdef JointClass < matlab.mixin.SetGet
   % JointClass is a class for generating the parameters and methods defining a
   % viscoelastic joint between two given beads. In the event there is only one
   % body in the system, the joint will be between that body and the ground.
   %
   %  Author: Abiy Wubneh
   %  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
   %
   properties
      % STATIC VARIABLES
```

```
11        jointType = 'Viscoelastic joint';
12        bodies = {};
13        material = [];
14        param = [];
15        % Fiber length
16        linkOriginalLength = 0;
17        nozzleUnitVec = [];
18        pos = [];
19        collectorDistance = 0;
20        % DYNAMIC VARIABLES
21        stress = 0;
22     end
23     methods
24        %% CONSTRUCTOR
25        %function thisJoint = ViscoelasticJoint(stress,bodyPair,param)
26        function thisJoint = JointClass(bodyPair,param)
27           thisJoint.param = param;
28           thisJoint.bodies = bodyPair;
29           thisJoint.material = param.material;
30           thisJoint.collectorDistance = param.equipment.collectorDistance;
31           thisJoint.nozzleUnitVec = param.equipment.nozzleUnitVec;
32           thisJoint.linkOriginalLength = param.fiber.initialLength;
33        end
34        %%
35        function pos = jointPosFn(THISJOINT,t)
36           %jointFposFn identifies the joint connection points on each body it
37           % connects. If the number of bodies in the system is 1, then the first
38           % position is taken as the ground connection point, which, in this case,
39           % is the nozzle tip. If the number of bodies is 2, the points in question
40           % represent the center of mass of the beads where the joint is assumed to
41           % be connected to the bodies.
42           %
43           % THISJOINT is a joint object
44           % t is the dynamic time variable
45           if numel(THISJOINT.bodies) == 1
46              [pos1,~,~] = nozzlePosVelAccFcn(t,THISJOINT.param.gCode);
47              pos2 = THISJOINT.bodies{1}.pos;
48           elseif numel(THISJOINT.bodies) == 2
49              pos1 = THISJOINT.bodies{1}.pos;
50              pos2 = THISJOINT.bodies{2}.pos;
51           else
52              error('At one or two bodies are requered to form a joint!')
53           end
54           pos =  0.5*(pos1 + pos2);
55        end
56        %%
57        function linkLength = linkCurrentLengthFn(THISJOINT,t)
58           %linkCurrentLengthFn measures the instantaneous distance between the two
59           % bodies THISJOINT is connecting.
60           %
61           if numel(THISJOINT.bodies) == 1
62              [pos1,~,~] = nozzlePosVelAccFcn(t, THISJOINT.param.gCode);
63              pos2 = THISJOINT.bodies{1}.pos;
64           elseif numel(THISJOINT.bodies) == 2
```

```
65            pos1 = THISJOINT.bodies{1}.pos;
66            pos2 = THISJOINT.bodies{2}.pos;
67         else
68            error('At one or two bodies are requered to form a joint!')
69         end
70         linkLength =  norm(pos2-pos1);
71      end
72      %%
73      function linkExtension = linkExtensionFn(THISJOINT,t)
74         %linkExtensionFn measures the amount of extension (or compression) taken
75         % place from the original gap between the two bodies THISJOINT is
76         % connecting.
77         %
78         linkExtension = THISJOINT.linkCurrentLengthFn(t) - ...
79            THISJOINT.linkOriginalLength;
80      end
81      %%
82      function unitVec = unitVectorFn(THISJOINT,t)
83         %unitVectorFn returns a unit vector pointing from the center of mass of
84         % the lower body (pos1) to the center of mass of the upper body (pos2)
85         % that THISJOINT connects.
86         %
87         if numel(THISJOINT.bodies) == 1
88            [pos1,~,~] = nozzlePosVelAccFcn(t, THISJOINT.param.gCode);
89            pos2 = THISJOINT.bodies{1}.pos;
90         elseif numel(THISJOINT.bodies) == 2
91            pos1 = THISJOINT.bodies{1}.pos;
92            pos2 = THISJOINT.bodies{2}.pos;
93         else
94            error('At one or two bodies are requered to form a joint!')
95         end
96         if norm(pos2-pos1) == 0
97            unitVec = -THISJOINT.nozzleUnitVec;
98         else
99            unitVec = (pos2-pos1)/norm(pos2-pos1);
100           % unit vector from Point 1 to Point 2
101        end
102     end
103     %%
104     function relativeVelocity = relativeVelocityFn(THISJOINT,t)
105        % relativeVelocityFn(JOINT,t) returns the velocity of the upper body
106        % (Body 2) relative to the lower body (Body 1).
107        %
108        if numel(THISJOINT.bodies) == 1
109           [~,pos1_dt,~] = nozzlePosVelAccFcn(t, THISJOINT.param.gCode);
110           pos2_dt = THISJOINT.bodies{1}.pos_dt;
111        elseif numel(THISJOINT.bodies) == 2
112           pos1_dt = THISJOINT.bodies{1}.pos_dt;
113           pos2_dt = THISJOINT.bodies{2}.pos_dt;
114        else
115           error('At one or two bodies are requered to form a joint!')
116        end
117        relativeVelocity = pos2_dt - pos1_dt;
118     end
```

```
    %% Constraints
    function [A1,A2,b_baumgarte] = AbFn(THISJOINT,t,Alpha,Beta)
       % AbFn returns the coefficient matrix components [A1 A2] and vector b of
       % the viscoelastic joint constraint equation.
       %
       % The constraint equation, phi(t) = 0, its first time derivative
       % phi_dt(t) = 0, and second-time derivative phi_ddt(t) = 0 are defined so
       % that the Baumgarte numerical stabilization method is employed. i.e.,
       %     phi_ddt + 2*alpha*phi_dt + beta^2*phi = 0  (Eq. 1)
       % is used in the solution in place of the original
       %                  phi_ddt = 0.
       % The constraint equation in Eq.1 is rearranged to get the result in the
       %               A*q_ddt - b = 0
       % form, specifically partitioning the A matrix:
       %         [A1 A2] q_ddt - b = 0.
       % The deviation follows these steps:
       %
       bodyCount = numel(THISJOINT.bodies);
       if bodyCount == 1
          [pos1,pos1_dt,~] = nozzlePosVelAccFcn(t, THISJOINT.param.gCode);
          pos2 = THISJOINT.bodies{1}.pos;
          pos2_dt = THISJOINT.bodies{1}.pos_dt;
       elseif bodyCount == 2
          pos1 = THISJOINT.bodies{1}.pos;
          pos2 = THISJOINT.bodies{2}.pos;
          pos1_dt = THISJOINT.bodies{1}.pos_dt;
          pos2_dt = THISJOINT.bodies{2}.pos_dt;
       else
          error('At one or two bodies are requered to form a joint!')
       end
       pos_rel = pos2 - pos1;
       vel_rel = pos2_dt - pos1_dt;
       pos_rel_skew = [0 -pos_rel(1) -pos_rel(2) -pos_rel(3);
          pos_rel(1) 0 -pos_rel(3) pos_rel(2);
          pos_rel(2) pos_rel(3) 0 -pos_rel(1);
          pos_rel(3) -pos_rel(2) pos_rel(1) 0];
       R = THISJOINT.bodies{bodyCount}.RMatFn;
       E = THISJOINT.bodies{bodyCount}.EMatFn;
       E_dt = THISJOINT.bodies{bodyCount}.EdtMatFn;
       G_dt = THISJOINT.bodies{bodyCount}.GdtMatFn;
       % Constraint equations in matrix form (two equations!)
       holonomicConstraint = R.'*pos_rel;
       holonomicConstraint = holonomicConstraint(1:2);
       % First-time derivative of the above constraint equations
       holonomicConstraint_dt = R.'*vel_rel + 2*E*G_dt.'*pos_rel;
       holonomicConstraint_dt = holonomicConstraint_dt(1:2);
       % Second-time derivative of the constraints
       A1 = [-R.', zeros(3,4)];
       A2 = [R.', 2*E*pos_rel_skew];
       b = -4*E*G_dt.'*vel_rel - 2*E_dt*G_dt.'*pos_rel;
       if numel(THISJOINT.bodies) == 1
          [~,~,nozzleAcceleration] = nozzlePosVelAccFcn(t,THISJOINT.param.gCode);
          b = b - A1*[nozzleAcceleration; 0; 0; 0; 0];
          % the nozzle has constant orientation, hence zero angular velocity and
```

```
            % angular acceleration
            A1 = zeros(3,7);
         end
         b = b(1:2);
         A1 = A1(1:2,:);
         A2 = A2(1:2,:);
         % Baumgarte constraint stablization
         b_baumgarte = b-2*Alpha*holonomicConstraint_dt-Beta^2*holonomicConstraint;
         % Ref. [ 2 ]
      end
      %%
      function stress_dt = stress_dtFn(THISJOINT,t)
         % stress_dtFn returns the time derivative of the
         % tensile stress in the viscoelastic joint.
         jointPos = THISJOINT. jointPosFn(t);
         [E0, E1, eta1] = elasticParametersFcn(THISJOINT.material, ...
            THISJOINT.collectorDistance,  jointPos(3));
         relativeVelocity = THISJOINT.relativeVelocityFn(t);
         unitVector = THISJOINT.unitVectorFn(t);
         extensionVelocity = dot(relativeVelocity,unitVector)*unitVector;
         stress_dt = -(E1/eta1)*THISJOINT.stress  + ...
            (E0*E1/eta1)*THISJOINT.linkExtensionFn(t) + ...
            (E0+E1)*dot(extensionVelocity,unitVector);
      end
   end
end
```

Listing A.4: Multi-body class

```
classdef MultibodyClass
   % MultibodyClass creates an instances of a multibody system baed on number of
   % active beads in the system. It handles all interactions between beads,
   % including forming equation of motions and andling constraints.
   %
   %  Author: Abiy Wubneh
   %  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
   %
   properties
      N = 0; % Number of beads
      param = struct([]);
      bodies = {};
      joints = {};
      numOfActive = 0;
      tableTouchedRegister = [];
      t = 0;
      X = [];
      x0 = [];
      dim = [];
   end
   methods
      %% CONSTRUCTOR
      function thisMultibody = MultibodyClass(t, X, param, numOfActive, ...
            tableTouchedRegister, x0, dim)
         thisMultibody.numOfActive = numOfActive;
```

```
26          thisMultibody.tableTouchedRegister = tableTouchedRegister;
27          thisMultibody.param = param;
28          thisMultibody.bodies = generateBodiesFn(thisMultibody);
29          thisMultibody.joints = generateJointsFn(thisMultibody);
30          thisMultibody.t = t;
31          thisMultibody.X = X;
32          thisMultibody.x0 = x0;
33          thisMultibody.dim = dim;
34       end
35       %% BODIES
36       function BODIES = generateBodiesFn(THISMULTIBODY)
37          % generateBodiesFn generates bodies in the multibody system based on the
38          % number of active bodies.
39          params = THISMULTIBODY.param;
40          BODIES = cell(THISMULTIBODY.numOfActive,1);
41          % list of bodies doesn't include the Ground link
42          for i = 1:THISMULTIBODY.numOfActive
43             BODIES{i} = BodyClass(params);
44          end
45       end
46       %% JOINTS
47       function JOINTS = generateJointsFn(THISMULTIBODY)
48          params = THISMULTIBODY.param;
49          BODIES = THISMULTIBODY.bodies;
50          JOINTS = cell(THISMULTIBODY.numOfActive,1);
51          % Each body has a joint connecting it to the previuos body. In the case of
52          % only one body, the joint connects the body to the ground.
53          for i = 1:THISMULTIBODY.numOfActive
54             if i == 1
55                bodyPair = BODIES(i);
56                % note: {A{i}} is same as A(1), but the latter is even faster
57             elseif i > 1
58                bodyPair = {BODIES{i-1} BODIES{i}};
59             end
60             JOINTS{i} = JointClass(bodyPair,params);
61          end
62       end
63       %% MASS MATRIX
64       function [M_udd, M_uddInv, M_udd05, M_udd_05, M_ODE, M_ODEInv] = ...
65             massMatrixFn(THISMULTIBODY)
66          M_ODE = eye(15*THISMULTIBODY.numOfActive);
67          M_ODEInv = eye(15*THISMULTIBODY.numOfActive);
68          CofVec = cell(THISMULTIBODY.numOfActive,1);
69          CofVec_inv = cell(THISMULTIBODY.numOfActive,1);
70          CofVec_05 = cell(THISMULTIBODY.numOfActive,1);
71          CofVec__05 = cell(THISMULTIBODY.numOfActive,1);
72          CofVec(:) = {THISMULTIBODY.param.massMatrix.cofVec};
73          CofVec_inv(:) = {THISMULTIBODY.param.massMatrix.cofVec_inv};
74          CofVec_05(:) = {THISMULTIBODY.param.massMatrix.cofVec_05};
75          CofVec__05(:) = {THISMULTIBODY.param.massMatrix.cofVec__05};
76          M_udd = diag(cell2mat(CofVec));             % Coefficient matrix from EOM
77          M_uddInv = diag(cell2mat(CofVec_inv));
78          M_udd05 = diag(cell2mat(CofVec_05));
79          M_udd_05 = diag(cell2mat(CofVec__05));
```

```
        M_ODE(7*THISMULTIBODY.numOfActive+1:2*7*THISMULTIBODY.numOfActive, ...
           7*THISMULTIBODY.numOfActive+1:2*7*THISMULTIBODY.numOfActive) = M_udd;
        M_ODEInv(7*THISMULTIBODY.numOfActive+1:2*7*THISMULTIBODY.numOfActive,...
          7*THISMULTIBODY.numOfActive+1:2*7*THISMULTIBODY.numOfActive) = M_uddInv;
    end
    %% CONSTRAINTS
    function [A,b] = AbFn(THISMULTIBODY,t)
        %% global numOfActive tableTouchedRegister x0 dim
        alpha_baumgarte = THISMULTIBODY.param.solver.Baumgarte.alpha_baumgarte;
        beta_baumgarte = THISMULTIBODY.param.solver.Baumgarte.beta_baumgarte;
        %% Body constraints
        A_bodies = zeros(THISMULTIBODY.numOfActive, ...
           THISMULTIBODY.dim*THISMULTIBODY.numOfActive);
        b_bodies = zeros(THISMULTIBODY.numOfActive,1);
        for i = 1:THISMULTIBODY.numOfActive
           idx1 = i*THISMULTIBODY.dim-6;
           idx2 =i*THISMULTIBODY.dim;
           [A_bodies(i,idx1:idx2), b_bodies(i)] = ...
              THISMULTIBODY.bodies{i}.AbFn(alpha_baumgarte,beta_baumgarte);
        end
        %% Joint constraints
        A_joints = zeros(THISMULTIBODY.numOfActive*...
           2,THISMULTIBODY.dim*(THISMULTIBODY.numOfActive+1));
        % +1 because the ground is temporarily included in the matrix formation
        b_joints = zeros(THISMULTIBODY.numOfActive*2,1);
        for i = 1:THISMULTIBODY.numOfActive
           idx1 = i*2-1;
           idx2 = i*2;
           jdx1 = i*THISMULTIBODY.dim-6;
           jdx2 =i*THISMULTIBODY.dim;
           [A_joints(idx1:idx2,jdx1:jdx2), A_joints(idx1:idx2,...
              jdx1+THISMULTIBODY.dim:jdx2+THISMULTIBODY.dim), ...
              b_joints(idx1:idx2)] = ...
              THISMULTIBODY.joints{i}.AbFn(t,alpha_baumgarte,beta_baumgarte);
        end
        A_joints(:,1:THISMULTIBODY.dim) = [];
        %% Table-sticking constraints
        if ~any(THISMULTIBODY.tableTouchedRegister)
           % If there are no beads stuck on the collector yet, use empity matrices
           % for A and b.
           A_table = [];
           b_table_baumgarte = [];
        elseif any(THISMULTIBODY.tableTouchedRegister)
           % if any beads are touching the table, ...
           stuckBeadsIndx = find(THISMULTIBODY.tableTouchedRegister);
           %  find the indexes of these beads
           A_table = cell(numel(stuckBeadsIndx),THISMULTIBODY.numOfActive);
           % create a matrix to contain constraints for these stuck beads
           A_table(:) = {zeros(3,THISMULTIBODY.dim)};
           % fill the initial matirx with zeros
           for row = 1:numel(stuckBeadsIndx)
              indx = stuckBeadsIndx(row);
              A_table{row,indx} = [eye(3), zeros(3,4)];
           end
```

```
            A_table = cell2mat(A_table);
            b_table = zeros(3*numel(stuckBeadsIndx),1);
            posStates_Short = cell(numel(stuckBeadsIndx),1);
            % table_pos_states(:) = {zeros(3,1)};
            posStates_dt_Short = cell(numel(stuckBeadsIndx),1);
            posTargets_short = cell(numel(stuckBeadsIndx),1);
            posTargets_Full = ...
               (THISMULTIBODY.x0(1:THISMULTIBODY.dim*THISMULTIBODY.numOfActive))';
            for row = 1:numel(stuckBeadsIndx)
               indx = stuckBeadsIndx(row);
               r1 = THISMULTIBODY.dim*(indx-1)+1;
               r2 = r1+2;
               posStates_Short{row,1} = THISMULTIBODY.bodies{indx}.pos;
               posStates_dt_Short{row,1} = THISMULTIBODY.bodies{indx}.pos_dt;
               posTargets_short{row,1} = posTargets_Full(r1:r2);
            end
            posStates_Short = cell2mat(posStates_Short);
            posStates_dt_Short = cell2mat(posStates_dt_Short);
            posTargets_short = cell2mat(posTargets_short);
            holonomicConstraint = posStates_Short - posTargets_short;
            holonomicConstraint_dt = posStates_dt_Short;
            % - targetPos_dt_short;
            b_table_baumgarte = ...
               b_table - 2*alpha_baumgarte*holonomicConstraint_dt - ...
               beta_baumgarte^2*holonomicConstraint;
         end
         %% assembled A and b
         b = [b_bodies; b_joints; b_table_baumgarte];
         A = [A_bodies; A_joints; A_table];
      end
      %% FORCES
      %             function [Qtotal, M_ODEInv] = forcesFn(THISMULTIBODY,t)
      %% 1. Force of gravity and its moment
      function Q_gravity = forceGravityFn(THISMULTIBODY)
         % global numOfActive
         if THISMULTIBODY.param.forces.includeGravitationalForces == true
            Q_gravity = cell(THISMULTIBODY.numOfActive,1);
            Q_gravity(:) = {THISMULTIBODY.param.forces.gravityForce};
            Q_gravity = cell2mat(Q_gravity);
         else
            Q_gravity = zeros(THISMULTIBODY.numOfActive*7,1);
         end
      end
      %% Electric field forces
      function Q_electric = forceElectricFieldFn(THISMULTIBODY)
         if THISMULTIBODY.param.forces.includeElectricFieldForces == true
            Q_electric =  cell(THISMULTIBODY.numOfActive,1);
            Q_electric(:) = {THISMULTIBODY.param.forces.electricFieldForce};
            Q_electric = cell2mat(Q_electric);
         else
            Q_electric =  zeros(THISMULTIBODY.numOfActive*7,1);
         end
      end
      %% Coulomb forces
```

```
188       function Q_coulomb = forceCoulomb(THISMULTIBODY,t)
189          ke = 8.987E09;
190          % Otion 1 ... runs slower than Option 2
191          if THISMULTIBODY.param.forces.includeCoulombForces == 0
192             Q_coulomb = zeros(THISMULTIBODY.numOfActive*7,1);
193          else
194             beadCharge =  THISMULTIBODY.param.bead.charge;
195             POS_vec_i = ...
196                cell(THISMULTIBODY.numOfActive+1,THISMULTIBODY.numOfActive+1);
197             [nozPos, ~, ~] = nozzlePosVelAccFcn(t, THISMULTIBODY.param.gCode);
198             POS_vec_i(1,:) = {[nozPos; zeros(4,1)]};
199             % It is assumed that the electric charge on the nozzle itself is
200             % contributing to the repelling of the beads away from the nozzle
201             for i = 1:THISMULTIBODY.numOfActive
202                POS_vec_i(i+1,:) = {[THISMULTIBODY.bodies{i}.pos; zeros(4,1)]};
203             end
204             Rji = cellfun(@minus,transpose(POS_vec_i), ...
205                POS_vec_i,'UniformOutput',false);
206             Q_coulomb = cellfun(@(x) - ...
207                ke*beadCharge^2*x/(norm(x))^3, Rji, 'UniformOutput',false);
208             Q_coulomb(1,:) = [];
209             Q_coulomb = cell2mat( Q_coulomb);
210             Q_coulomb(isnan(Q_coulomb)) = 0;
211             Q_coulomb = sum(Q_coulomb,2);
212          end
213       end
214       %% Gyroscopic moments
215       function Q_gyroscopic = momentGyroscopicFn(THISMULTIBODY)
216          % global numOfActive
217          Q_gyroscopic = zeros(THISMULTIBODY.numOfActive*7,1);
218          for i = 1:THISMULTIBODY.numOfActive
219             idx1 = i*7-3;
220             idx2 = idx1+3;
221             Q_gyroscopic(idx1:idx2) = ...
222                - 4*THISMULTIBODY.param.bead.Jo*...
223                (norm(THISMULTIBODY.bodies{i}.quat_dt))^2*....
224                THISMULTIBODY.bodies{i}.quat;
225          end
226       end
227       %% Joint viscoelastic forces
228       function Q_jointsLinear = forceViscoelasticFn(THISMULTIBODY,t)
229          Q_jointsLinear = ...
230             zeros((THISMULTIBODY.numOfActive+1)*7, THISMULTIBODY.numOfActive);
231          if THISMULTIBODY.param.forces.includeViscoelasticLinear == true
232             for i = 1:THISMULTIBODY.numOfActive
233                idx1 = i*7-6;
234                idx2 = idx1+2;
235                fiberCrosssectionalArea = ...
236                   THISMULTIBODY.param.bead.volume/...
237                   THISMULTIBODY.joints{i}.linkCurrentLengthFn(t); % m^2
238                FveOnLowerBody = ...
239                   fiberCrosssectionalArea*THISMULTIBODY.joints{i}.stress*...
240                   THISMULTIBODY.joints{i}.unitVectorFn(t);
241                % Viscoelastic force applied on the ground/nozzle
```

```
            FveOnThisBody = -FveOnLowerBody;
            Q_jointsLinear(idx1:idx2,i) = FveOnLowerBody;
            Q_jointsLinear(idx1+7:idx2+7,i) = FveOnThisBody;
         end
      end
      Q_jointsLinear(1:7,:) = [];
      Q_jointsLinear = sum(Q_jointsLinear,2);
   end
   %% Joint viscoelastic moments
   function Q_jointTorques = momentViscoelasticFn(THISMULTIBODY,t)
      Q_jointTorques = ...
         zeros((THISMULTIBODY.numOfActive+1)*7, THISMULTIBODY.numOfActive);
      if THISMULTIBODY.param.forces.includeViscoelasticTorsional == true
         for i = 1:THISMULTIBODY.numOfActive
            idx1 = i*7-3;     % from q0 to q3
            idx2 = idx1+3;
            if numel(THISMULTIBODY.joints{i}.bodies) == 1
               u_lowerBody = THISMULTIBODY.param.equipment.nozzleUnitVec;
               u_thisBody = THISMULTIBODY.joints{i}.unitVectorFn(t);
               G_lowerBody = ...
                  quatGMatrixFcn(THISMULTIBODY.param.equipment.nozzleQuat);
               G_thisBody = THISMULTIBODY.joints{i}.bodies{1}.GMatFn;
               N_omeg1_n = [0; 0; 0];
               [N_omeg2_n,~] = THISMULTIBODY.joints{i}.bodies{1}.omegFn;
               %z_position = THISMULTIBODY.joints{i}.bodies{1}.pos(3);
            elseif numel(THISMULTIBODY.joints{i}.bodies) == 2
               u_lowerBody = THISMULTIBODY.joints{i-1}.unitVectorFn(t);
               u_thisBody = THISMULTIBODY.joints{i}.unitVectorFn(t);
               G_lowerBody = THISMULTIBODY.joints{i}.bodies{1}.GMatFn;
               G_thisBody = THISMULTIBODY.joints{i}.bodies{2}.GMatFn;
               [N_omeg1_n,~] = THISMULTIBODY.joints{i}.bodies{1}.omegFn;
               [N_omeg2_n,~] = THISMULTIBODY.joints{i}.bodies{2}.omegFn;
               %z_position = THISMULTIBODY.joints{i}.bodies{2}.pos(3);
            end
            jointPos = THISMULTIBODY.joints{i}.jointPosFn(t);
            Kt = KtFcn(THISMULTIBODY.param.material, ...
               THISMULTIBODY.param.equipment.collectorDistance, jointPos(3));
            Ct = CtFcn(THISMULTIBODY.param.material, ...
               THISMULTIBODY.param.equipment.collectorDistance, jointPos(3));
            unitVecsCrossed = cross(u_lowerBody,u_thisBody);
            unitVecsVrossed_Norm = norm(unitVecsCrossed);
            includedAngle = ...
               atan2(unitVecsVrossed_Norm,dot(u_lowerBody,u_thisBody));

            if unitVecsVrossed_Norm == 0
               hingeUnitVec = [1; 0; 0];
            else
               hingeUnitVec = unitVecsCrossed/unitVecsVrossed_Norm;
            end
            N_omeg21_n = N_omeg2_n - N_omeg1_n;
            % angular velocity of this body relative to the lower body
            N_TorqueOnThisBody_n = ...
               -Kt*includedAngle*hingeUnitVec - Ct*N_omeg21_n;
            N_TorqueOnLowerBody_n = - N_TorqueOnThisBody_n;
```

```
296                 Q_jointTorques(idx1+7:idx2+7,i) = ...
297                    2*G_thisBody.'*N_TorqueOnThisBody_n;
298                 Q_jointTorques(idx1:idx2,i) = 2*G_lowerBody.'*N_TorqueOnLowerBody_n;
299                 % 2*E_T*N_Gamma_n
300              end
301           end
302           Q_jointTorques(1:7,:) = [];
303           % this represents the reaction torque on the nozzle applied by the exiting
304           % bead
305           Q_jointTorques = sum(Q_jointTorques,2);
306        end
307        %% Q_total
308        function [Qtotal, M_ODEInv] = forcesFn(THISMULTIBODY,t)
309           Q = THISMULTIBODY.forceGravityFn + ...
310              THISMULTIBODY.forceElectricFieldFn + ...
311              THISMULTIBODY.forceCoulomb(t) + ...
312              THISMULTIBODY.momentGyroscopicFn + ...
313              THISMULTIBODY.forceViscoelasticFn(t) + ...
314              THISMULTIBODY.momentViscoelasticFn(t);
315           [A,b] = THISMULTIBODY.AbFn(t);
316           [~, M_uddInv, M_udd05, M_udd_05, ~, M_ODEInv] = ...
317              THISMULTIBODY.massMatrixFn;
318           Q_constraint = M_udd05*pinv(A*M_udd_05)*(b - A*M_uddInv*Q);
319           Qtotal = Q + Q_constraint;
320        end
321        %% EQUATION OF MOTION
322        function [Udd] = EOMFn(THISMULTIBODY)
323           POSQUAT = THISMULTIBODY.X(1:7*THISMULTIBODY.numOfActive,1);
324           POSQUAT_dt = THISMULTIBODY.X(7*THISMULTIBODY.numOfActive+1:2*7*...
325              THISMULTIBODY.numOfActive,1);
326           FVE = THISMULTIBODY.X(2*7*THISMULTIBODY.numOfActive+1:end,1);
327           u_dt = POSQUAT_dt;
328           for i = 1:THISMULTIBODY.numOfActive
329              idx1 = 7*i-6;
330              idx2 = 7*i;
331              X_posquat_i = POSQUAT(idx1:idx2);
332              X_posquat_dt_i = POSQUAT_dt(idx1:idx2);
333              THISMULTIBODY.bodies{i}.pos = X_posquat_i(1:3);
334              THISMULTIBODY.bodies{i}.quat = X_posquat_i(4:7);
335              THISMULTIBODY.bodies{i}.pos_dt =  X_posquat_dt_i(1:3);
336              THISMULTIBODY.bodies{i}.quat_dt =  X_posquat_dt_i(4:7);
337              THISMULTIBODY.joints{i}.stress = FVE(i);
338           end
339           stress_dt = zeros(THISMULTIBODY.numOfActive,1);
340           for i = 1:THISMULTIBODY.numOfActive
341              stress_dt(i) = THISMULTIBODY.joints{i}.stress_dtFn(THISMULTIBODY.t);
342           end
343           [Qtotal,M_ODEInv] = THISMULTIBODY.forcesFn(THISMULTIBODY.t);
344           Udd = M_ODEInv*[u_dt; Qtotal; stress_dt];
345        end
346     end
347  end
```

A.3 Functions

Listing A.5: allNaNEliminatorFcn: All-NaN rows or columns eliminator function

```
function cleanedMatrices = allNaNEliminatorFcn(varargin)
% ========================================================================================
% ALLNANELIMINATORFCN Eliminates rows columns or rows from input matrices
% based on all-NaN rows or columns in a reference matrix. If the ith row or
% column in the reference matrix is all NaN elements, and the corresponding
% rows or columns in the target matrices will be removed.
%
%  Input arguments:
%     varargin{1}: Selection option for row operations. If yes, the
%     function eliminates rows with all NaN elements
%     varargin{2}: Selection option for column operations. If yes, the
%     function eliminates rows with all NaN columns.
%     varargin{3}: Option to resize the target matrice to match the
%     reference matrix
%     varargin{4}: Reference matrix
%     varargin{5,6, ...}: Target matrices to be cleaned based on the
%     reference matrix.
%
%  Output argument:
%     CLEANEDMATRICES: A structure containing the cleaned versions of the
%     target and reference matrices. The members of the structure have the same
%     name as the input. The cleaned matrices could be accessed by using
%     the dot operation, i.e.,
%        member1 = CLEANEDMATRICES.member1
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta

referenceMatrix = varargin{4};
[ref_row, ref_col] = size(referenceMatrix);
clean_row = varargin{1};
clean_col = varargin{2};
free_form = varargin{3};
for i =4:nargin
   targetMatrix = varargin{i};
   [target_row, target_col] = size(targetMatrix);
   if free_form == yes
      if target_row > ref_row
         target_row = ref_row;
      end
      if target_col > ref_col
         target_col = ref_col;
      end
   elseif free_form == no
      if (clean_row == yes) && (target_row ~= ref_row)
         error(['Target ' num2str(i-3) ...
            ' and the reference matrices have incompatible row sizes!'])
      end
      if (clean_col == yes) && (target_col ~= ref_col)
         error(['Target ' num2str(i-3) ...
            ' and the reference matrices have incompatible column sizes!'])
```

```
      end
   end
   if varargin{1} == yes  % Remove blank rows based on reference matrix
      targetMatrix(~any(~isnan(referenceMatrix(1:target_row,:)),2),:) = [];
   end
   if varargin{2} == yes % Remove blank columns based on reference matrix
      targetMatrix(:,~any(~isnan(referenceMatrix(:,1:target_col)),1)) = [];
   end
   cleanedMatrices.(inputname(i)) = targetMatrix;
end
end
```

Listing A.6: animateFcn: Function for animating simulation results

```
function animateFcn(solnDirectoryPath, animationOption)
% ANIMATEFCN Animates the simulation result. It takes the solution path and
% animation options.
%
%  Input arguments
%     Input 1: Path to the directory where the solution is saved
%     Input 2: Animation option. If the value is yes, the animation of the
%     solution runs and saves a video file in the solution directory. If no, only
%     the final accumulation snapshot is returned as a single image. The image is
%     also saved in the same directory.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta

%% Import data
simSummaryPath = fullfile(solnDirectoryPath, 'SimulationSummary.mat');
tempSummaryStruct = load(simSummaryPath, 'simulationSummary');
simSummary = tempSummaryStruct.simulationSummary;
fiberChar = simSummary.fiberChar;
param = simSummary.param;
numOfSegements = param.solver.numOfSegments;
segArraySizesMatrix = param.solver.segSolArraySizes;
solnNumOfRows = sum(segArraySizesMatrix(:,1),1);
solnlNumofCol = segArraySizesMatrix(1,2);
rowLimit = 10000;
colLimit = 50000;
rowJump = ceil(solnNumOfRows/rowLimit);
animationNumOfRows = ceil(solnNumOfRows/rowJump) + numOfSegements;
tSol = NaN(animationNumOfRows, 1);
xSolNaN = NaN(animationNumOfRows, segArraySizesMatrix(1,2));
ySolNaN = NaN(animationNumOfRows, segArraySizesMatrix(1,2));
zSolNaN = NaN(animationNumOfRows, segArraySizesMatrix(1,2));
nozzlePos = NaN(animationNumOfRows, 3);
if lower(animationOption) == yes
    startingSegID = 1;
    startIndx = 1;
    for segID = startingSegID:1:numOfSegements
        fullFileName = ...
           fullfile(solnDirectoryPath, sprintf('SegmentSoln%d.mat',segID));
        tempSolStruct = load(fullFileName, 'segSol');
```

```
        soln = tempSolStruct.segSol;
        animSegLength = length(soln.time_vec(1:rowJump:end));
        endIndx = startIndx + animSegLength - 1;
        tSol(startIndx:endIndx,:) = soln.time_vec(1:rowJump:end);
        xSolNaN(startIndx:endIndx,:) = soln.x(1:rowJump:end,:);
        ySolNaN(startIndx:endIndx,:) = soln.y(1:rowJump:end,:);
        zSolNaN(startIndx:endIndx,:) = soln.z(1:rowJump:end,:);
        nozzlePos(startIndx:endIndx,:) = [soln.nozzlePos.x(1:rowJump:end,:), ...
           soln.nozzlePos.y(1:rowJump:end,:), soln.nozzlePos.z(1:rowJump:end,:)];
        clearvars soln tempSolStruct
        startIndx = endIndx + 1;
    end
else
    fullFileName = ...
       fullfile(solnDirectoryPath, sprintf('SegmentSoln%d.mat',numOfSegements));
    tempSolStruct = load(fullFileName, 'segSol');
    soln = tempSolStruct.segSol;
    tSol = soln.time_vec(end); % read only the last row of the last segment
    xSolNaN = soln.x(end,:);
    ySolNaN = soln.y(end,:);
    zSolNaN = soln.z(end,:);
    nozzlePos(1,:) = ...
       [soln.nozzlePos.x(end,:), soln.nozzlePos.y(end,:), soln.nozzlePos.z(end,:)];

    clearvars soln tempSolStruct
end
fullNaNRowRemoved = ...
   allNaNEliminatorFcn(yes,no,no,tSol,xSolNaN,ySolNaN,zSolNaN);
tSol = fullNaNRowRemoved.tSol;
xSolmm = 1000*fullNaNRowRemoved.xSolNaN;
ySolmm = 1000*fullNaNRowRemoved.ySolNaN;
zSolmm = 1000*fullNaNRowRemoved.zSolNaN;
nozzlePos = 1000*nozzlePos;
%% Check the quality of the solution data before plotting
% check if there are any NaN values exist in the solution
xNaNCheck = NaNCheckFcn(xSolmm);
yNaNCheck = NaNCheckFcn(ySolmm);
zNaNCheck = NaNCheckFcn(zSolmm);
if xNaNCheck || yNaNCheck || zNaNCheck
    error('NaN found!')
end
% Check if the dimensions of the solution data are compatible/correct/
if ~isequal(size(xSolmm), size(ySolmm), size(zSolmm))
    error('x, y, and z position solution matrices are not of euqal dimensions!')
end
[totalNumofTimeSeteps, numOfColumns] = size(xSolmm);
if totalNumofTimeSeteps ~= size(xSolmm,1)
    error('Length of solution time vector not same as the rest of the soluton!')
end
%% Create a smooth spline for smooth printing.
%posVecSpacedSmooth = smoothdata(posVecSpaced, 'gaussian', 150);
xSpline = cell(totalNumofTimeSeteps,1);
ySpline = cell(totalNumofTimeSeteps,1);
zSpline = cell(totalNumofTimeSeteps,1);
```

```
columnFiller = floor(colLimit/solnlNumofCol);
if columnFiller < 1
    columnFiller = 1;
elseif columnFiller > 8
    columnFiller = 8;
end
col = 1:numOfColumns;
col_linSpaced = linspace(1,numOfColumns, columnFiller*numOfColumns);
if animationOption == yes
    startingTimeIndx = 1;
else
    startingTimeIndx = totalNumofTimeSeteps;
end
for timeStep = startingTimeIndx:totalNumofTimeSeteps
    pp = spline(col,[xSolmm(timeStep,:); ySolmm(timeStep,:); zSolmm(timeStep,:)]);
    xyz_interpolated = ppval(pp,col_linSpaced);
    xSpline{timeStep} = xyz_interpolated(1,:);
    ySpline{timeStep} = xyz_interpolated(2,:);
    zSpline{timeStep} = xyz_interpolated(3,:);
end
xSpline = cell2mat(xSpline);
ySpline = cell2mat(ySpline);
zSpline = cell2mat(zSpline);
%% Start plotting
if lower(param.sessionInfo.isNozzleMoving) == yes
    xAxisLim = [0, param.equipment_setting.platformSize(1,1)];
    yAxisLim = [0, param.equipment_setting.platformSize(1,2)];
else
    xAxisLim = [min(xSpline,[],'all'), max(xSpline,[],'all')];
    yAxisLim = [min(ySpline,[],'all'), max(ySpline,[],'all')];
end
% ========================================================================
% If animation is required, the following runs.
if lower(animationOption) == yes
    nozzleColor = '#706b6b';
    nozzleMarkerSize = 15;
    nozzleOffset = nozzleMarkerSize/2 - 3;
    markerVector = nozzlePos;
    markerVector(:,3) = markerVector(:,3) + nozzleOffset;
    zAxisLim = [-param.equipment_setting.collectordistance-1, 0+2*nozzleOffset];

    Setting = param.equipmentSetting;
    inputParameterStr = {
        MATERIAL;
        Material name:  +param.material.materialName;
         ;
        PROCESSING PARAMETERS;
        Flow rate:  +num2str(Setting.flowrate)+ ml/hr;
        Applied voltage:  +num2str(Setting.voltage)+ kV;
        Collector distance:  +num2str(Setting.collectordistance)+ mm;
        Melt temperature:  +num2str(Setting.meltTemp)+ deg C;
         ;
        NOZZLE SPEED;
        Feed rate:   + num2str(param.gCode.feedRateOveride) +  mm/min};
```

```
    col1 = [0, 0.4470, 0.7410];
    annotationBackgroundColor = 1/255*[246 246 248];
    figure('Name',[simSummary.param.sessionInfo.name ...
       ': Animaton perspective view'], 'NumberTitle','off','Color','white');

    figHandele1 = gcf;
    if lower(param.sessionInfo.isNozzleMoving) == yes
        figHandele1.Position = [500 50 2.5*figHandele1.Position(3:4)];
    else
        figHandele1.Position = ...
           [500 50 1.75*figHandele1.Position(3) 2*figHandele1.Position(4)];
    end
    h_nozzlePath = plot3(nozzlePos(:,1), nozzlePos(:,2), nozzlePos(:,3),...
       ':', 'LineWidth',1.5,'color','red');
    hold('on')
    grid('on');
    axis equal;
    zlabel('Collector distance (mm)','FontSize',9,'FontWeight','normal');
    h_nozzleTip = plot3(markerVector(:,1), markerVector(:,2), markerVector(:,3),...
       ':', 'LineWidth',1.5,'Marker','v',MarkerSize=nozzleMarkerSize, ...
       MarkerFaceColor=nozzleColor, MarkerEdgeColor=nozzleColor);
    legend({'Nozzle path'},'Location','northeast');
    dimAnnot = [0.1 0.501 0.325 0.41];
    annotation('textbox',dimAnnot,'String',inputParameterStr,'FitBoxToText',...
       'on','FontSize',8,'BackgroundColor',annotationBackgroundColor);
    axisLimit = [xAxisLim yAxisLim zAxisLim];
    axis(axisLimit)
    clear timeStep
    for timeStep = 1:totalNumofTimeSeteps
        if timeStep == 1
            h_3Dfiber = ...
               plot3(xSpline(timeStep,:), ySpline(timeStep,:), ...
               zSpline(timeStep,:),'color',col1, 'LineWidth',0.25,...
               'DisplayName','Melt-electrospun fiber');
            view([30,25]);
            h_3Dtitle = title(sprintf('Time: %0.2f s',tSol(1)),...
               FontSize,10,FontWeight,normal);
        else
            set(h_3Dfiber, 'xdata', xSpline(timeStep,:), 'ydata', ...
               ySpline(timeStep,:), 'zdata', zSpline(timeStep,:))
            set(h_nozzlePath, 'xdata', nozzlePos(1:timeStep,1), 'ydata', ...
               nozzlePos(1:timeStep,2), 'zdata', nozzlePos(1:timeStep,3))
            set(h_nozzleTip, 'xdata', markerVector(timeStep,1), 'ydata', ...
               markerVector(timeStep,2), 'zdata', markerVector(timeStep,3))
            set(h_3Dtitle,'String',sprintf('t = %0.2f s',tSol(timeStep)))
        end
        frame_escape = 12;
        animationFileName = fullfile(solnDirectoryPath, 'Animation.gif');
        if timeStep == 1
            thisFrame = getframe(gcf);
            im = frame2im(thisFrame);
            [imind,cm] = rgb2ind(im,256);
            imwrite(imind,cm,animationFileName,'gif','DelayTime',0,...
               'Loopcount',inf);
```

```
        elseif mod(timeStep,frame_escape) == 0
            thisFrame = getframe(gcf);
            im = frame2im(thisFrame);
            [imind,cm] = rgb2ind(im,256);
            imwrite(imind,cm,animationFileName,'gif','DelayTime',0,...
               'WriteMode','append');
        end
    end
    figure1FileName = fullfile(solnDirectoryPath, 'Animation_Perspective.fig');
    savefig(figure1FileName);
    hold off
end
% Whether animation is requested or not, the function generates a 2D plot
inputParameterStr2 = {
    MATERIAL;
    Material name:  +param.material.materialName;
     ;
    PROCESSING PARAMETERS;
    Flow rate:   + ...
    num2str(param.equipmentSetting.flowrate) +  ml/hr;
    Applied voltage:   + num2str(param.equipmentSetting.voltage) +  kV;
    Collector distance:   + ...
    num2str(param.equipmentSetting.collectordistance)+ mm;
    Melt temperature:   + ...
    num2str(param.equipmentSetting.meltTemp)+ deg C;
     ;
    MODEL OUTPUT PARAMETERS;
    Collection diameter:  + ...
    num2str(1000*fiberChar.maximumCollectionDiameter,'%.1f')+ mm;
    Fiber diameter:   + ...
    num2str(1000000*fiberChar.averageFiberDiameter,'%.1f')+ };
axisLimit = [xAxisLim yAxisLim];
snapfig = figure ('Name', [simSummary.param.sessionInfo.name ...
   ': Animaton top view'], 'NumberTitle','off','Color','white');
% ========================================================================
figHandele2 = gcf;
if lower(param.sessionInfo.isNozzleMoving) == yes
    figHandele2.Position = [500 50 2.5*figHandele2.Position(3:4)];
end
col1 = [0, 0.4470, 0.7410];
p=get(axes,'Position');
h_2Dfiber = plot(xSpline(end,:), ySpline(end,:),'color',col1, 'LineWidth',0.25,...
   'DisplayName','Melt-electrospun fiber');
hold on
grid on
if lower(param.sessionInfo.isNozzleMoving) == yes
    plot(nozzlePos(:,1), nozzlePos(:,2),':', 'LineWidth',1.5,'color','red');
    lgnd = legend({'Melt-electrospun fiber','Nozzle path'},'Location','northeast');
    annotationString = {['Feed rate: ' ...
       num2str(simSummary.param.gCode.feedRateOveride) ' mm/min']};
    axis equal padded
    axis(axisLimit)
    boxDim = [p(1)+0.045, p(2)+0.01, 0.15, 0.025];
    annotation('textbox',boxDim,'String',annotationString,...
```

```
       'BackgroundColor',white);
else
    axis image
end
xlabel('Printer x-axis (mm)');
ylabel('Printer y-axis (mm)');
h_2Dtitle = title(sprintf('Time: %0.2f s',tSol(totalNumofTimeSeteps)),...
   FontSize,10,FontWeight,normal);
figure2FileName = fullfile(solnDirectoryPath, 'Animation_Top.fig');
savefig(figure2FileName);
end
```

Listing A.7: CtFcn: Torsional damping coefficient calculator

```
function Ct_z = CtFcn(material, H, z)
% CTFCN returns the torsional damping of the fiber for a given material property and
% distance from the nozzle. This function is currently designed to be a
% placeholder and needs to be updated in the future
%
%  Input arguments:
%     Input 1: Structure containing material property data
%     Input 2: Collector distance
%     Input 3: Current height variable
%
%  Output argument:
%     Output 1: Calculated torsional damping coefficient
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta

   C1 = 1.25;
   y = C1^(1000*z)-1;
   y_max = C1^(1000*H)-1;
   Ct_z = ((material.scaleCt - 1)*(y/y_max) + 1)*material.Ctt;
end
```

Listing A.8: ejectionTimesFcn: Time step and ejection time calculator

```
function [timeStep, ejectionTimeStep] = ejectionTimesFcn(equipment, solver, bead)
% EJECTIONTIMESFCN Calculates the numerical time step and the interval at
% which beads are ejected into the multi-body system.
%
%  Input arguments:
%     Input 1: A data structure containing details about the equipment
%     setting
%     Input 2: A data structure containing solver settings
%     Input 3: A data structure containing bead information
%
%  Output arguments
%     Output 1: Numerical time step
%     Output 2: Bead ejection time interval
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
```

```
%
volumericFlowrate = equipment.volumeric_flowrate;
nozzleDiameter = equipment.nozzleDiameter;
numOfTimeStepsPerEjection = solver.numOfStepsPerEjection;
slendernessRatio1 = bead.slendernessRatio1;
ejectionTimeStep = (slendernessRatio1*pi*nozzleDiameter^3)/(4*volumericFlowrate);
timeStep = ejectionTimeStep/numOfTimeStepsPerEjection;
timeStep = round(timeStep,5);
ejectionTimeStep = timeStep*numOfTimeStepsPerEjection;
end
```

Listing A.9: elasticParametersFcn.m

```
function [E0_z, E1_z, eta_z] = elasticParametersFcn(material, H, z_cord )
% ELASTICPARAMETERSFCN Calculates the elastic parameter values E_0, E_1,
% and eta at a given distance from the nozzle. These parameters are
% expected to change values as the fiber travels away from the nozzle and
% cools down to the chamber temperature.
% The function, in its current form, is very simplistic and mostly designed
% to serve as a modular placeholder for future improvements.
%
%  Input arguments:
%     Input 1: Data structure containing material information
%     Input 2: Collector distance
%     Input 3: The current z coordinate at which the elastic parameters are
%     calculated
%
%  Output arguments:
%     Output 1: Elastic parameter 1, E_0
%     Output 2: Elastic parameter 2, E_1
%     Output 3: Elastic Parameter 3, eta
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%
C1 = 1.25;
z = abs(z_cord);
E0_z = ((z/H)*(material.ambientTempFactor.E0 - 1) + 1)*material.E0t;
E1_z = ((z/H)*(material.ambientTempFactor.E1 - 1) + 1)*material.E1t;
eta_z = ((z/H)*(material.ambientTempFactor.eta - 1) + 1)*material.etat;
end
```

Listing A.10: euler2MatrixFcn.m

```
function R = euler2MatrixFcn(eul)
%EULER2MATRIXFCN Generates rotational matrix from a given set of Euler
%angles.
%
%  Input arguments:
%     Input 1: An array of three Euler angles
%
%  Output argument:
%     Output 1: Rotational matrix R
%
```

```
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%
   Rz = [cos(eul(1)) -sin(eul(1)) 0;
       sin(eul(1)) cos(eul(1)) 0;
       0 0 1];
   Ry = [cos(eul(2)) 0 sin(eul(2));
       0 1 0;
       -sin(eul(2)) 0 cos(eul(2))];
   Rx = [1 0 0;
       0 cos(eul(3)) -sin(eul(3));
       0 sin(eul(3)) cos(eul(3))];
   R = Rz*Ry*Rx;
end
```

Listing A.11: euler2QuatFcn

```
function q = euler2QuatFcn(angles,rotationSsequence)
%EULER2QUATFCN takes in three Euler angles and their sequence and returns
% a unit quaternion representing the same rotation.
%
%  Input arguments (angles,rotationSsequence)
%     angles: a three-length vector containing the three Euler angles in
%     radians.
%     rotationSequence: a string containing the sequence of rotation. This
%     can be written either in a letter form (e.g., 'ZXY or 'zyz') or as a
%     sequence of numbers (e.g., '321')
%
%  Output argument
%     [q]: a four-length vector representing the quaternion.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%

%% Check for dimensional accuracy of input parameters
if length(angles) ~= 3
   error('The length of Euler angles vector must be equal to 3!')
end
%% Parse variables
psi = angles(1);
theta = angles(2);
phi = angles(3);
%% Calculate quaternion
rotationSsequence = lower(rotationSsequence);
switch rotationSsequence
   case {'zyx','321',''}
      qh = [cos(psi/2); sin(psi/2)*[0 0 1]'];
      qa = [cos(theta/2); sin(theta/2)*[0 1 0]'];
      qb = [cos(phi/2); sin(phi/2)*[1 0 0]'];
   case {'zxz','313'}
      qh = [cos(psi/2); sin(psi/2)*[0 0 1]'];
      qa = [cos(theta/2); sin(theta/2)*[1 0 0]'];
      qb = [cos(phi/2); sin(phi/2)*[0 0 1]'];
```

```
   case {'zyz','323'}
      qh = [cos(psi/2); sin(psi/2)*[0 0 1]'];
      qa = [cos(theta/2); sin(theta/2)*[0 1 0]'];
      qb = [cos(phi/2); sin(phi/2)*[0 0 1]'];
   case{'xyz','123'}
      qh = [cos(psi/2); sin(psi/2)*[1 0 0]'];
      qa = [cos(theta/2); sin(theta/2)*[0 1 0]'];
      qb = [cos(phi/2); sin(phi/2)*[0 0 1]'];
   otherwise
      error('u:stuffed:it', ['Unrecognized rotation sequence entered. \n' ...
         'This sequence might not have been added to the program yet. \n' ...
         'Update the code and try again.'])
end
qab = quatProdFcn(qa,qb);
q = quatProdFcn(qh,qab);
end
```

Listing A.12: fiberCharactersticsFcn

```
function  fiberChar = fiberCharactersticsFcn(soln)
%FIBERCHARACTERSTICSFCN Analyzes a simulation result and calculates the average
% fiber diameters and the collection diameters.
%
%  Input arguments
%     Input 1: Data structure contacting solution data
%
%  Output argument
%     Output 1: Data structure containing the calculated fiber diameter and
%     collection diameter
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%

initialFiberDiameter = soln.param.fiber.initialDiameter;
initialLinkLength = soln.param.fiber.initialLength;
T = soln.time_vec;
x = soln.x;
y = soln.y;
z = soln.z;
x_dot = soln.x_dot;
y_dot = soln.y_dot;
z_dot = soln.z_dot;
xp = soln.nozzlePos.x;
yp = soln.nozzlePos.y;
zp = soln.nozzlePos.z;
clear soln
[R,C] = size(x);
linkLength = zeros(R,C);
linkDiameter = zeros(R,C);
collectionRadius = zeros(R,C);
beadSpeed = zeros(R,C);
for row = 1:R
   for column = 1:C
```

```
      if column == 1
         disp_ij = [x(row,column) - xp(row), y(row,column) - yp(row), ...
            z(row,column) - zp(row)];
      else
         disp_ij = [x(row,column) - x(row,column-1), y(row,column) - ...
            y(row,column-1), z(row,column) - z(row,column-1)];
      end
      radPos_ij = [x(row,column), y(row,column)];
      speed_ij = [x_dot(row,column), y_dot(row,column), z_dot(row,column)];
      instantaneousLinkLength = norm(disp_ij);
      linkLength(row,column) = instantaneousLinkLength;
      linkDiameter(row,column) = ...
         initialFiberDiameter*sqrt(initialLinkLength/instantaneousLinkLength);
      collectionRadius(row,column) = norm(radPos_ij);
      beadSpeed(row,column) = norm(speed_ij);
   end
end
linkDiameter(isinf(linkDiameter)|linkDiameter == 0) = NaN;
collectionRadius(collectionRadius == 0) = NaN;
fiberChar.averageFiberDiameter = mean(linkDiameter,all,'omitnan');
fiberChar.maximumCollectionDiameter = 2*max(collectionRadius,[],all,'omitnan');
fiberChar.max_beadSpeed = max(max(beadSpeed));
fiberChar.percentage_elongation = ...
   100*(max(max(linkLength)) - initialLinkLength)/initialLinkLength;
end
```

Listing A.13: gCodeKinematicsFcn

```
function gCode_struct = gCodeKinematicsFcn(gCode_struct)
%GCODEKINEMATICSFCN imports and reads a g-code file. It reconstructs the g-code
% inside MATLAB. In addition, it calculates the position, velocity, and
% acceleration profiles of the nozzle based on the g-code.
%
%  Input arguments
%     Input 1: Data structure contacting g-code path and setting values
%
%  Output argument
%     Output 1: The g-code data structure, updated with the kinematic profiles.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%

%% Part 1: Import and reconstruct the G-code
gcodeTextFile = fopen(gCode_struct.filePath);
globalReadData = {};
lineReadData = {'' 0 0 0 0 0 0};
while ~feof(gcodeTextFile)
   lineString = fgetl(gcodeTextFile);
   if ~isempty(lineString) && lineString(1) == 'G'
      if strcmp(lineString(1:3),'G20')
         gCode_struct.unit = inch;
      elseif strcmp(lineString(1:3),'G21')
         gCode_struct.unit = mm;
```

```
      else
         % Next, check if there is any comment in the line
         commentIndx = strfind(lineString, '(');
         lineString(commentIndx:end) = [];
         subStrings = split(lineString);
         [numOfSubstrings, ~] = size(subStrings);
         for subStringID = 1:numOfSubstrings
            % get the marker identifying symbol, .i.e., G, X Y, Z, I, J, or F
            IDLetter = subStrings{subStringID}(1);
            switch IDLetter
               case 'G'
                  lineReadData{1} = subStrings{subStringID};
                  if strcmp(subStrings{subStringID},'G00')
                     lineReadData{7} = gCode_struct.rapidOveride;
                  end
               case 'X'
                  lineReadData{2} = str2double(subStrings{subStringID}(2:end));
               case 'Y'
                  lineReadData{3} = str2double(subStrings{subStringID}(2:end));
               case 'Z'
                  lineReadData{4} = 0.0; % nozle z position overide.
               case 'I'
                  lineReadData{5} = str2double(subStrings{subStringID}(2:end));
               case 'J'
                  lineReadData{6} = str2double(subStrings{subStringID}(2:end));
               case 'F'
                  if ~isempty(gCode_struct.feedRateOveride)
                     lineReadData{7} = gCode_struct.feedRateOveride;
                  else
                     lineReadData{7} = str2double(subStrings{subStringID}(2:end));
                  end
            end
         end
         globalReadData = [globalReadData; lineReadData];
         lineReadData{5} = 0;
         lineReadData{6} = 0;
      end
   end
end
gCode_struct.gCodeTable = lineReadData;
%{
   At this point, the g-code file is imported and reconstructed with slight
   modifications. The original rapid motion and normal feed rate values are
   overwritten by the new values specified by the user in the dashboard.
%}
%% Part 2:  Interpolate positions and build the time profile of the path
[numOfsubPathes, ~] = size(globalReadData);
timePosFeed = [0 0 0 0 0];   % [time x y z Feedrate]
for gCodeLineNum = 1:numOfsubPathes
   gNum = globalReadData(gCodeLineNum,1);
   if gCodeLineNum == 1
      initialPos = [0,0,0]; % [Time x y z]
   elseif gCodeLineNum > 1
      initialPos = cell2mat(globalReadData(gCodeLineNum-1,2:4));
```

```
    end
    commandedPos_cell = globalReadData(gCodeLineNum,:);
    tf_previous = timePosFeed(end,1);
    timePosFeedCurrent = ...
        lineInterpolatorFcn(commandedPos_cell, initialPos, gCode_struct.unit);
    timePosFeedCurrent(:,1) = timePosFeedCurrent(:,1) + tf_previous;
    timePosFeed = [timePosFeed; timePosFeedCurrent];
end
%% Adjusting spacing
%{
    At this point, we have the time vector and the interpolated position vector.
    Column 1 is the time vector, and Column 2,3,4 are the x, y, and z are values of
    the position vector.

    The time spacing is not uniform!

    Make sure the position data is smoothened before calculating its velocity and
    acceleration. Otherwise, it could result in spikes and outliers in the velocity
    and acceleration curves which cause lots of problems during the numerical
integration stages.
%}
[~,ia,~] = unique(timePosFeed(:,1));
timePosTrimmed = timePosFeed(ia,:);
frontIndx = find(timePosTrimmed(:,5) == gCode_struct.feedRateOveride,1);
backIndx = find(timePosTrimmed(:,5) == gCode_struct.feedRateOveride,1,last);
timePosTrimmed = timePosTrimmed(frontIndx:backIndx,1:4);
timePosTrimmed(:,1) = timePosTrimmed(:,1) - timePosTrimmed(1,1); % time starts at 0
timePosTrimmed(:,4) = 0;  % mute motion of nozzle along the z axis
timeRaw = timePosTrimmed(:,1);
posRaw = timePosTrimmed(:,2:4);
timeStepsRaw = NaN(numel(timeRaw)-1,1);
for i = 1:numel(timeRaw)-1
    timeStepsRaw(i,1) = timeRaw(i+1,1) - timeRaw(i,1);
end
gaussianValue = 2;
tr_Feedrate = 10;
timeStep_unudjested = tr_Feedrate*mean(timeStepsRaw,'all');
numOfTimeSteps = ceil(timeRaw(end)/timeStep_unudjested);
timeVecSpaced = (linspace(0,timeRaw(end),numOfTimeSteps)).';
timeStep = timeRaw(end)/(numOfTimeSteps-1);
posVecSpaced = interp1(timeRaw,posRaw,timeVecSpaced);
posVecSpacedSmooth = smoothdata(posVecSpaced, 'gaussian', gaussianValue);
%% Part 3: Calculate velocity and accelerations profiles
numRow = numel(timeVecSpaced);
velVec = NaN(numRow,3);
accVec = NaN(numRow,3);
velVec(1,:) = [0 0 0];
accVec(1,:) = [0 0 0];
for divID = 2:numRow
    dispVecCurrent = posVecSpacedSmooth(divID,:) - posVecSpacedSmooth(divID-1,:);
    velVec(divID,:) = dispVecCurrent/timeStep;
end
velVecSpaced = smoothdata(velVec, 'gaussian',1);
for divID = 2:numRow
```

```
    velVecCurrent = velVec(divID,:) - velVec(divID-1,:);
    accVec(divID,:) = velVecCurrent/timeStep;
end
accVecSpaced = smoothdata(accVec, 'gaussian',1);
interpolatedKinmatics = ...
    [timeVecSpaced, posVecSpacedSmooth, velVecSpaced, accVecSpaced];
gCode_struct.totalTime = interpolatedKinmatics(end,1);
%% Unit Conversion to m (i.e.,  mm -> m   or in -- m)
%{
    Everything up until this point is calculated based on the measuring units
    imported with the G-code. The final array has to be converted into a meters (m)
    unit before being returened by the function.
%}
if gCode_struct.unit == mm
    posVelAcc = interpolatedKinmatics(:,2:end)/1000;
elseif gCode_struct.unit == inch
    posVelAcc.interpolatedKinmatics = interpolatedKinmatics(:,2:end)*(25.4/1000);
end
gCode_struct.gCodeKinematicsData.t = interpolatedKinmatics(:,1);
gCode_struct.gCodeKinematicsData.pos = posVelAcc(:,1:3);
gCode_struct.gCodeKinematicsData.vel = posVelAcc(:,4:6);
gCode_struct.gCodeKinematicsData.acc = posVelAcc(:,7:9);
end
```

Listing A.14: gCodeplotForInspectionFcn

```
function  figure_gCode = gCodeplotForInspectionFcn(param)
%GCODEPLOTFORINSPECTIONFCN Plots the g-code reconstructed in MATLAB. It also plots
% the kinematic profiles of the nozzle, including position, velocity, and
% acceleration as a function of time steps that match that of the numerical
% integration time steps.
%
%  Input arguments
%     Input 1: Data structure contacting g-code data and calculated
%     kinematic profiles
%
%  Output argument
%     Output 1: Figure handle of the plots
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL @ Mechanical Engineering, University of Alberta
%

gCode = param.gCode;
xAxisLim = [0, param.equipment_setting.platformSize(1,1)];
yAxisLim = [0, param.equipment_setting.platformSize(1,2)];
axisLimit = [xAxisLim yAxisLim];
t = gCode.gCodeKinematicsData.t;
p = gCode.gCodeKinematicsData.pos;
v = gCode.gCodeKinematicsData.vel;
a = gCode.gCodeKinematicsData.acc;
figure_gCode = figure('Name', [param.sessionInfo.name ...
    ': G-code kinematics profile (Feed rate: ' ...
    num2str(param.gCode.feedRateOveride) ' mm/min)'], 'NumberTitle','off');
```

```
figHandele = gcf;
figHandele.Position = [817 120 2*figHandele.Position(3:4)];
tiled01 = tiledlayout(2,2);
xColor = '#4279bd';
yColor = '#bd3422';

% Tile 1
nexttile
plot(1000*p(:,1), 1000*p(:,2), LineWidth=1, Color='k')
axis equal padded
title('Nozzle path imported from G-code')
xlabel('x position (mm)');
ylabel('y positoin (mm)');
axis(axisLimit)

% Tile 2
nexttile
plot(t,p(:,1),LineWidth=1,Color=xColor)
hold on
plot(t,p(:,2),LineWidth=1,Color=yColor)
axis padded
title('Nozzle position vs. time')
legend('x position','y position')
xlabel('Time (s)')
ylabel('Position (m)')
hold off

% Tile 3
nexttile
plot(t,v(:,1),LineWidth=1,Color=xColor)
hold on
plot(t,v(:,2),LineWidth=1,Color=yColor)
hold off
axis padded
title('Nozzle velocity vs. time')
legend('x velocity','y velocity')
xlabel('Time (s)')
ylabel('Nozzle velocity (m/s)')

% Tile 4
nexttile
plot(t,a(:,1),LineWidth=1,Color=xColor)
hold on
plot(t,a(:,2),LineWidth=1,Color=yColor)
axis padded
title('Nozzle acceleration vs. time')
legend('x acceleration','y acceleration')
xlabel('Time (s)')
ylabel('Nozzle acceleration (m/s^2)')
hold off
tiled01.Padding = 'compact';
tiled01.TileSpacing = 'compact';
end
```

Listing A.15: initialConditionFcn

```
function InitialCondition = initialConditionFcn(lastSol,t, param, numOfActive, ...
   tableTouchedRegister, lastActiveIndx, dim)
%INITIALCONDITIONFCN Generates initial conditions for iterations. It reads the last
%simulation results and passes them as initial values for the subsequent iterations
%
%  Input arguments
%     Input 1: Solution of the last iteration
%     Input 2: Current simulation time
%     Input 3: Data structure containing parameter values
%     Input 4: Number of active beads in the current iteration
%     Input 5: A list of bead indexes touching the table
%     Input 6: The index of the last active bead in the current iteration
%     Input 7: Number of independent generalized coordinates per bead
%
%  Output argument
%     Output 1: Generated initial conditions for all beads in the current
%     iteration
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

unstretchedLinkLength = param.fiber.initialLength;
H = param.equipment.collectorDistance;
[~,E1,~] = elasticParametersFcn(param.material,H,0);

%% Nozzle kinematics
[nozPos, nozPos_dt, ~] = nozzlePosVelAccFcn(t, param.gCode);
nozzleQuat = param.equipment.nozzleQuat; % assumed constant during simulation

%% Randomized quaternions => position
posPurturbationScale = 0.01; % percentage of the free link length (no purturbation)

zEulerRandom = (-360 + (360-(-360)).*rand(1,1)); % random angle [-360 and 360] deg
yEulerRandom = (-15 + (15-(-15)).*rand(1,1));    % random angle [-15 and 15] deg
xEulerRandom = 0;

ZYXrad = deg2rad([zEulerRandom, yEulerRandom, xEulerRandom]);
Noz_quat_exitBead = euler2QuatFcn(ZYXrad ,'zyx');
quat_exitBead = quatProdFcn2(nozzleQuat,Noz_quat_exitBead);
unitVec_exitBead = quat2RotMatrixFcn(quat_exitBead)*param.env.inertial_z_axis;
B_pos_exitBead_n = posPurturbationScale*unstretchedLinkLength*unitVec_exitBead;
posQuat_exitBead = [nozPos + B_pos_exitBead_n; quat_exitBead];

%% Randomized quaternions and quaternions derivatives => postition derivatives

quat_dt_exitBead = ...
   quatExpFcn(quatLogFcn(quatDivideFcn(nozzleQuat,quat_exitBead))/...
   (param.solver.ejectionTimeStep));
N_omega_n = 2*quatProdFcn2(quat_dt_exitBead,quatConjugateFcn(quat_exitBead));
pos_dt_exitBead = nozPos_dt + cross(N_omega_n(2:end), B_pos_exitBead_n) + ...
   param.bead.exitSpeed*param.equipment.nozzleUnitVec;
```

```
posQuat_dt_exitBead = [pos_dt_exitBead;quat_dt_exitBead];

%% Estimate the viscoelastic force variable
Fve_exitBead = E1*(0-unstretchedLinkLength);
%{
normally, F = E1*((Pos2-Pos1)- unstretchedLinkLength). But since all
points insdie the nozzle are considered to have the nozzle's kinematic
profile, Pos1 = Pos2.
%}

%% Use the solution of the last iteration as an initial condition for the next

if isempty(lastSol)
    posQuat_fromLastSol =[];
    posQuat_dt_fromLastSol = [];
    fve_dt_fromLastSol = [];
else
    posQuat_fromLastSol = lastSol(1:dim*lastActiveIndx);
    posQuat_dt_fromLastSol = ...
       lastSol(dim*numOfActive+1:dim*(numOfActive + lastActiveIndx));
    fve_dt_fromLastSol = ...
       lastSol(2*dim*numOfActive+1: 2*dim*numOfActive+lastActiveIndx);

    if any(tableTouchedRegister)
        stuckBeadIndexes = find(tableTouchedRegister);
        for i = stuckBeadIndexes(1):lastActiveIndx
            ix = (i-1)*dim + 1;
            iz = ix + 2;
            posQuat_dt_fromLastSol(ix:iz) = [0, 0, 0];
        end
    end
end

%% Assemble the new initial condition vector
InitialCondition = [posQuat_exitBead', posQuat_fromLastSol, ...
    posQuat_dt_exitBead',posQuat_dt_fromLastSol, ...
    Fve_exitBead,fve_dt_fromLastSol];
end
```

Listing A.16: KtFcn: Torsional stiffness constant

```
function Kt_z = KtFcn(material, H, z)
%KTFCN Calculates the torsional elastic constant for the material at z distance
% from the nozzle. This function is too simplistic to do a good job, and is only
% designed to be a modular placeholder for future develpments.
%
%  Input arguments
%     Input 1: Data structure containing material parameters
%     Input 2: Collector distance
%     Input 3: Current z coordinate value at which the elastic parameter is
%     calculated for
%
%  Output argument
%     Output 1: Calculated torsional elastic constant
```

```
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
   C1 = 1.25;
   y = C1^(1000*z)-1;
   y_max = C1^(1000*H)-1;
   Kt_z = ((material.scaleKt - 1)*(y/y_max) + 1)*material.Ktt;
end
```

Listing A.17: lineInterpolatorFcn

```
function tPosFeed = lineInterpolatorFcn(commandedPos_cellArray, initialPos, unit)

gNumber = commandedPos_cellArray{1};
commandedPos = cell2mat(commandedPos_cellArray(1,2:end));

startXYZ = initialPos;
endXYZ = commandedPos(1,1:3);
offsetXYZ = [commandedPos(4:5), 0];
feedRate = [commandedPos(1,6)];
travelSpeed = feedRate/60;

tPosFeed = [];

currentTime = 0;
previousPos = startXYZ;
lineDiv = 0.5; % mm

%% If linear motion ...

if strcmp(gNumber,'G00') || strcmp(gNumber,'G01')

    if lower(unit) == mm
        linearDivisionDistance = lineDiv;
    elseif lower(unit) == inch
        linearDivisionDistance = lineDiv/25.4;
    else
        error('Unknown unit!')
    end

    commandedDisplacement = endXYZ - startXYZ;
    commandedDistance = norm(commandedDisplacement);
    divisionTravelTime = linearDivisionDistance/travelSpeed;

    if commandedDistance == 0
        divisionFraction = 0;
    else
        divisionFraction = linearDivisionDistance/commandedDistance;
    end

    currentDistance = 0;
    k = 1;
    while currentDistance <= commandedDistance
```

```
        tPosFeed(k,:) = [currentTime, previousPos];
        previousPos = previousPos + divisionFraction*commandedDisplacement;
        currentTime = currentTime + divisionTravelTime;
        currentDistance = currentDistance + linearDivisionDistance;
        k = k+1;
    end
    tPosFeed(end,2:4) = endXYZ;
    clear k

end

%% If arc motions ...

if strcmp(gNumber,'G02') || strcmp(gNumber,'G03')

    angleDivisionDeg = 0.5; % degrees

    centerXY = startXYZ + offsetXYZ;
    startLineVec = startXYZ - centerXY;
    endLineVec = endXYZ - centerXY;

    radius = norm(endLineVec);

    startAngle = vectorAngleFcn(startLineVec,d);
    endAngle = vectorAngleFcn(endLineVec,d);

    if strcmp(gNumber, 'G03')  % Counterclockwise

        if startAngle > endAngle
            startAngle = startAngle - 360;
        end

        currentAngle = startAngle;
        i = 1;
        while currentAngle <= endAngle
            calculatedPos = centerXY + radius*[cosd(currentAngle), sind(currentAngle)
                , 0];
            tPosFeed(i,:) = [currentTime, calculatedPos];
            divisionDistanceTravelled = norm(calculatedPos - previousPos);
            currentTime = currentTime + divisionDistanceTravelled/travelSpeed;
            currentAngle = currentAngle + angleDivisionDeg;
            previousPos = calculatedPos;
            i = i+1;
        end

    elseif strcmp(gNumber,'G02') % clockwise

        if startAngle < endAngle
            startAngle = startAngle + 360;
        end

        currentAngle = startAngle;
        i = 1;
        while currentAngle >= endAngle
```

```
            calculatedPos = centerXY + radius*[cosd(currentAngle), sind(currentAngle)
                , 0];
            tPosFeed(i,:) = [currentTime, calculatedPos];
            divisionDistanceTravelled = norm(calculatedPos - previousPos);
            currentTime = currentTime + divisionDistanceTravelled/travelSpeed;
            currentAngle = currentAngle - angleDivisionDeg;
            previousPos = calculatedPos;
            i = i+1;
        end
    end
    tPosFeed(end,2:4) = endXYZ;

end

tPosFeed = [tPosFeed, feedRate*ones(size(tPosFeed,1),1)];
end
```

Listing A.18: lineNaNCheckerFcn

```
function [rowCheckStatus, colCheckStatus] = lineNaNCheckerFcn(matrixToBeChecked)
%LINENANCHECKERFCN Checks if there is at least one row or column with all NaN
% elements. The function returns logical 1 in output index position 1 if there
% exists at least one row with all NaN elements and 0 if three are none. Output
% index position 2 is a similar check status for any columns in the given
% matrix with all NaN elements.
%
%  Input arguments
%     Input 1: Array/matrix to be checked for NAN elements
%
%  Output argument
%     Output 1: Row check status
%     Output 2: Column check status
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
   [rowSize, colSize] = size(matrixToBeChecked);
   matrixToBeChecked(~isnan(matrixToBeChecked)) = 0;
   matrixToBeChecked(isnan(matrixToBeChecked)) = 1;
   rowSum = sum(matrixToBeChecked, 2);
   colSum = sum(matrixToBeChecked, 1);
   rowCheckStatus = any(rowSum == colSize);
   colCheckStatus = any(colSum == rowSize);
end
```

Listing A.19: NaNCheckFcn: NaN values checking function

```
function checkStatus = NaNCheckFcn(matrixToBeChecked)
%NANCHECKFCN Checks if there are NAN elements in the input matrix
%
%  Input arguments
%     Input 1: Matrix to be checked for NAN
%
%  Output argument
```

```
%     Output 1: Check status
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
   matrixToBeChecked(~isnan(matrixToBeChecked)) = 0;
   matrixToBeChecked(isnan(matrixToBeChecked)) = 1;
   sumTotal = sum(sum(matrixToBeChecked));
   checkStatus = sumTotal >= 1;
end
```

Listing A.20: niceTableDisplayFcn

```
function out = niceTableDisplayFcn(tbl)
%NICETABLEDISPLAYFCN Displays tabular text in a readable format.
%
%  Input argument:
%     Input 1: Tabular data to be displayed
%
%  Out rgument:
%     No returened argument
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
out = tbl;
for i = 1:width(tbl)
    if iscellstr(tbl{:,i}) || isstring(tbl{:,i})
        out.(out.Properties.VariableNames{i}) = categorical(tbl{:,i});
    end
end
end
```

Listing A.21: nozzlePosVelAccFcn

```
function [nozzlePos,nozzleVel,nozzleAcc] = nozzlePosVelAccFcn(varargin)
%NOZZLEPOSVELACCFCN Calculates and returns the position, velocity, and acceleration
% vectors at each acquisition point are vertical vectors, i.e., e.g.,
% pos(t) = [x_t; y_t; z_t]. However, the same vectors in the imported kinematic
% data are horizontal vectors. Appropriate conversion from one form to another is
% essential.
%
%  Input arguments
%     Input 1: Arry of time
%     Input 2: G-code data, available only if the nozzle is motiving
%
%  Output argument
%     Output 1: Nozzle position calculated at the acquisition time
%     Output 2: Sampled nozzle velocity
%     Output 3: Acceleration
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
```

```
t_q = varargin{1};
if nargin == 1
   gCode = [];
elseif nargin == 2
   gCode = varargin{2};
end
if isempty(gCode.gCodeKinematicsData)
   nozzlePos = zeros(3, numel(t_q));
   nozzleVel = zeros(3, numel(t_q));
   nozzleAcc = zeros(3, numel(t_q));
else
   gCodeData_tq = ...
      (interp1(gCode.gCodeKinematicsData.t, [gCode.gCodeKinematicsData.pos, ...
      gCode.gCodeKinematicsData.vel, gCode.gCodeKinematicsData.acc], t_q)).';
   nozzlePos = gCodeData_tq(1:3,:);
   nozzleVel = gCodeData_tq(4:6,:);
   nozzleAcc = gCodeData_tq(7:9,:);
   if sum(isnan(nozzlePos),'all') > 0
      gPosEnd = (gCode.gCodeKinematicsData.pos(end,:)).';
      numOfCol = size(nozzlePos,2);
      firstNaNIndx = find(isnan(nozzlePos(1,:)),1);
      nozzlePos(:,firstNaNIndx:end) = repmat(gPosEnd, 1, numOfCol-firstNaNIndx+1);
      nozzleVel(:,firstNaNIndx:end) = zeros(3, numOfCol-firstNaNIndx+1);
      nozzleAcc(:,firstNaNIndx:end) = zeros(3, numOfCol-firstNaNIndx+1);
   end
end
end
```

Listing A.22: ParameterFcn: Parameters manager function

```
function param = ParametersFcn(equipment_setting, solver, forces, material)
%PARAMETERSFCN Assembles parameter values and process setting data into a data
% structure. The data is needed by the solver and other function %downstream. It
% is the single most important file since it contains all information needed to
% run the simulation.
%
%  Input arguments
%     Input 1: Equipment setting values, such as temp, flowrate, voltage,
%     Input 2: Data structure containing solver setting
%     Input 3: Data structure containing force parameters and options
%     Input 4: Data structure containing material-related data
%
%  Output argument
%     Output 1: An updated parameter data structure updated with additional
%     information
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

%% Environment
env.gravityVec = [0; 0; -9.81];     % (m/s^2)
env.inertial_z_axis = [0; 0; 1];

```

```
%% Equipment Setting
equipment.volumeric_flowrate_ml_per_hr = equipment_setting.flowrate;      % (ml/hr)
equipment.volumeric_flowrate = ...
   equipment.volumeric_flowrate_ml_per_hr*2.777777778E-10;    % (m^3/s)
equipment.voltage = equipment_setting.voltage*1000;                       % (volt)
equipment.meltTemperature = equipment_setting.meltTemp;                   % (deg C)
equipment.oven_temperature = equipment_setting.ambTemp;                   % (deg C)
equipment.collectorDistance = equipment_setting.collectordistance/1000;   % (m)
equipment.nozzleDiameter = equipment_setting.nozzleDiameter/1000;         % (m)
equipment.nozzleEulerAnglesZYX = ...
   deg2rad([0 0 180 + equipment_setting.directionOfSpinning]);
equipment.nozzleQuat = euler2QuatFcn(equipment.nozzleEulerAnglesZYX ,'zyx');
equipment.nozzleUnitVec = ....
   euler2MatrixFcn(equipment.nozzleEulerAnglesZYX)*env.inertial_z_axis;
equipment.collectorPlateUnitVec = - equipment.nozzleUnitVec;
equipment.electricFieldVector = equipment.nozzleUnitVec;

%% Material Properties
material.materialName = 'PLA';
material.density = 1073;      % (kg/m^3)
material.Ktt = 50E-14;        % N-m/rad
material.Ctt = 1E-14;         % (2E-03)

%% Bead properties (assumed a sphere) (2022)
bead.bodyType = 'Viscoelastic element';
bead.shape = 'Shphere';
bead.slendernessRatio1 = 1*equipment.volumeric_flowrate_ml_per_hr;
bead.slendernessRatio2 = 3;
bead.radius = equipment.nozzleDiameter*(3*bead.slendernessRatio1/16)^(1/3);% (m)
bead.volume = (4/3)*pi*bead.radius^3;                       % ... assuming sphere
bead.mass = material.density*bead.volume;                   % (kg)
bead.surfaceArea = 4*pi*bead.radius^2;                      % (m^2)
bead.Jo = (2/5)*bead.mass*bead.radius^2;                    % = (2/5*m*r^2);
bead.surfaceChargeDensity = 2.9333E-05;                     % (C/m^2)
bead.charge = bead.surfaceArea*bead.surfaceChargeDensity;   %(C)
bead.exitSpeed = 4*equipment.volumeric_flowrate/(pi*equipment.nozzleDiameter^2);

%% Fiber segment properties (assumed cylinderical) (2022)
fiber.initialDiameter = ...
   equipment.nozzleDiameter*(bead.slendernessRatio1/bead.slendernessRatio2)^(1/3);
fiber.initialLength = equipment.nozzleDiameter*bead.slendernessRatio2*...
   (bead.slendernessRatio1/bead.slendernessRatio2)^(1/3);
fiber.shape = 'cylinder';

%% Matrix
massMatrix.cofVec = [bead.mass*[1;1;1]; 4*bead.Jo*[1;1;1;1]];
massMatrix.cofVec_inv = massMatrix.cofVec.^-1;
massMatrix.cofVec_05 = massMatrix.cofVec.^0.5;
massMatrix.cofVec__05 = massMatrix.cofVec.^-0.5;

%% Forces
forces.electricFieldForce = ...
   [bead.charge*(equipment.voltage/equipment.collectorDistance)*...
   equipment.electricFieldVector; zeros(4,1)];
```

```
% Force = charge*Electric_field = charge * (Voltage/distance) * unitVec
forces.gravityForce = [bead.mass*env.gravityVec; zeros(4,1)];

%% Solver Setting
[timeStep, ejectionTimeSetp] = ejectionTimesFcn(equipment, solver, bead);
solver.timeStep =timeStep;
solver.ejectionTimeStep = ejectionTimeSetp;
solver.Baumgarte = solver.Baumgarte;
solver.stuckBeadLimit = solver.stuckBeadLimit;

%% Assemble above data into a single structure
param.env = env;
param.equipmentSetting = equipment_setting;
param.equipment = equipment;
param.material = material;
param.fiber = fiber;
param.bead = bead;
param.massMatrix = massMatrix;
param.forces = forces;
param.solver = solver;

save('ParameterFile','param')
clearvars -except param
end
```

Listing A.23: plotForInspectionFcn

```
function  plotForInspectionFcn(soln, segmentID, param)
%PLOTFORINSPECTION Plots simulation results at the end of each simulation segment.
%
%  Input arguments
%     Input 1: Data structure containing segment solution data
%     Input 2: ID of a segment whose solution is being inspected
%     Input 3: Data structure containing solution parameter data
%
%  Output argument
%     Output 1: Plots with subfigures showing the same segment solution
%     from different point of views
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
solutionPath = param.solver.solutionPath;
xSolmm = 1000*soln.x;
ySolmm = 1000*soln.y;
zSolmm = 1000*soln.z;
clear soln
figure('Name', [param.sessionInfo.name ': Segment ' num2str(segmentID) ...
   ' --- End of segment simulation inspection'], 'NumberTitle','off');
figHandele = gcf;
figHandele.Position = [817 120 2*figHandele.Position(3:4)];
tiled01 = tiledlayout(2,2);

% Tile 1
```

```
nexttile
plot(xSolmm, zSolmm,'LineWidth',1.25);
axis padded
title('Font view')
xlabel('x position (mm)');
ylabel('z positoin (mm)');
xlim auto
ylim auto

% Tile 2
nexttile
plot(ySolmm, zSolmm,'LineWidth',1.25);
axis padded
title('Side view');
xlabel('y position (mm)');
ylabel('z position (mm)');
xlim auto
ylim auto

% Tile 3
nexttile
plot(xSolmm, ySolmm,'LineWidth',1.25)
axis padded
title('Top view');
xlabel('x position (mm)');
ylabel('y position (mm)');
xlim auto
ylim auto

% Tile 4
nexttile
plot3(xSolmm, ySolmm, zSolmm, 'LineWidth',1.25)
axis padded
title('Perspective view');
xlabel('x position (mm)');
ylabel('y position (mm)');
zlabel('z position (mm)');
xlim auto
ylim auto
zlim auto
tiled01.Padding = 'compact';
tiled01.TileSpacing = 'compact';
savefig([solutionPath '/SegementInspectioinPlots' num2str(segmentID) '.fig']);
end
```

Listing A.24: positionsFillerFcn

```
function [X_MAT,Y_MAT,Z_MAT] = positionsFillerFcn(X_MAT,Y_MAT,Z_MAT,nozzlePos)
%POSITIONFILLERFCN Fills NaN and zeros elements in the current segment solution
%matrices with appropriate values.
%
%  Input arguments
%     Input 1: x position solution matrix
%     Input 2: y position solution matrix
```

```
%     Input 3: z position solution matrix
%     Input 4: the nozzle position vector
%
%  Output argument
%     Output 1: Filled x position matrix
%     Output 2: Filled y position matrix
%     Output 3: Filled z-position matrix
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

   [~,numOfColumns] = size(X_MAT);
   for i = 1:numOfColumns
      lastNonNaNIndx = find(~isnan(X_MAT(:,i)),1,'last');
      X_MAT(lastNonNaNIndx:end,i) = X_MAT(lastNonNaNIndx,i);
      Y_MAT(lastNonNaNIndx:end,i) = Y_MAT(lastNonNaNIndx,i);
      Z_MAT(lastNonNaNIndx:end,i) = Z_MAT(lastNonNaNIndx,i);
      lastNaNIndx = find(isnan(X_MAT(:,i)),1,'last');
      X_MAT(1:lastNaNIndx,i) = nozzlePos(1:lastNaNIndx,1);
      Y_MAT(1:lastNaNIndx,i) = nozzlePos(1:lastNaNIndx,2);
      Z_MAT(1:lastNaNIndx,i) = nozzlePos(1:lastNaNIndx,3);
   end
end
```

Listing A.25: quat2RotMatrixFcn

```
function RotationMatrix = quat2RotMatrixFcn(q)
%QUAT2ROTMATRIXFCN Generates a rotation matrix from a given quaternion.
%
%  Input argument:
%     Input 1: Quaternion
%
%  Output argument:
%     Output 1: The calculated rotation matrix (3x3)
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

   RotationMatrix = ...
      [2*(q(1)^2+q(2)^2)-1, 2*(q(2)*q(3)-q(1)*q(4)), 2*(q(2)*q(4)+q(1)*q(3));
       2*(q(2)*q(3)+q(1)*q(4)), 2*(q(1)^2+q(3)^2)-1, 2*(q(3)*q(4)-q(1)*q(2));
       2*(q(2)*q(4)-q(1)*q(3)), 2*(q(3)*q(4)+q(1)*q(2)), 2*(q(1)^2+q(4)^2)-1];
end
```

Listing A.26: quatConjugateFcn

```
function qConjugate = quatConjugateFcn(q)
%QUATCONJGATEFCN Calculates the conjugate of a given quaternion.
%
%  Input argument:
%     Input 1: The quaternion whose conjugate is to be calculated
%
```

```
%  Output argument:
%     Output 1: Calculated quaternion conjugate
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

    %% check first the sizes of the input arguments
    if length(q) ~= 4
        error('The entered quaternions must be four-length vectors.')
    end
    %% calculate elements of the product of the two quaternions
    q(2:end) = -q(2:end);
    qConjugate = q;
end
```

Listing A.27: quatDivideFcn

```
function quatDivision = quatDivideFcn(q,r)
%QUATDIVIDERFCN Divides one quaternion by anoter quaternion
%
%  Input arguments:
%     Input 1: The first quaternion as a four-element vector
%     Input 2: The second quaternion as a four-element vector
%
%  Output argument:
%     Output: A quaternion
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
    quatDivision = zeros(size(q));
    r_norm_sqrd = sum(r.*r);
    quatDivision(1) = sum(q.*r)/r_norm_sqrd;
    quatDivision(2) = (r(1)*q(2) - r(2)*q(1) + r(4)*q(3))/r_norm_sqrd;
    quatDivision(3) = (r(1)*q(3) + r(2)*q(4) - r(3)*q(1) - r(4)*q(2))/r_norm_sqrd;
    quatDivision(4) = (r(1)*q(4) - r(2)*q(3) + r(3)*q(2) - r(4)*q(1))/r_norm_sqrd;
end
```

Listing A.28: quatExpFcn

```
function quatExp = quatExpFcn(q)
%QUATEXPFCN Calculates the exponent of a given quaternion
%
%  Input arguments:
%     Input 1: A quaternion as a four-element vector
%
%  Output argument:
%     Output: A quaternion as a four-element vector
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
    w = q(1);
```

```
   v = q(2:end);
   quatExp = zeros(size(q));
   quatExp(1) = exp(w)*cos(norm(v));
   quatExp(2:end) = exp(w)*(v/norm(v))*sin(norm(v));
end
```

Listing A.29: quatGMatrixFcn

```
function G = quatGMatrixFcn(q)
%QUAGMATRIXFCN Calculates Calculates the G-matrix froma given quaternion
%
%  Input arguments:
%     Input 1: A quaternion as a four-element vector
%
%  Output argument:
%     Output: A 3x3 matrix (G matrix)
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

G = [-q(2) q(1) -q(4) q(3);
      -q(3) q(4) q(1) -q(2);
      -q(4) -q(3) q(2) q(1)];
end
```

Listing A.30: quatLogFcn

```
function quatLog = quatLogFcn(q)
%QUATLOGFCN Calculates Calculates the logarithm of a given quaternion
%
%  Input arguments:
%     Input 1: A quaternion as a four-element vector
%
%  Output argument:
%     Output 1: A quaternion as a four-element vector
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

   a = q(1);
   v = q(2:end);
   quatLog = zeros(size(q));
   quatLog(1) = log(norm(q));
   quatLog(2:end) = (   v/(norm(v))   )    *acos(   a/(norm(q))    );
end
```

Listing A.31: quatProdFcn

```
function q1q2 = quatProdFcn(q1,q2)
%QUATPRODFCN Calculates the product of two quaternions. It takes two unit
% quaternions as input arguments and returns their product.
%
```

```
%  Input arguments: q1 and q2
%     The quaternions must be four-length row or column vectors or in a mixed
%     format (e.g., one could be a column vector and the other a row vector.)
%
%  Output argument: q1q2 = q1*q2
%     The output quaternion is also a four-length vector. It inherits its size
%     dimensions from the input argument, provided both inputs have similar sizes.
%     Otherwise, the output will be returned as a 4X1 column vector.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

    %% check first the sizes of the the input arguments
    if length(q1) ~= 4 || length(q2) ~= 4
        error('The entered quaternions must be four-length vectors.')
    end

    %% calculate elements of the product of the two quaternions
    n0 = q2(1)*q1(1) - q2(2)*q1(2) - q2(3)*q1(3) - q2(4)*q1(4);
    n1 = q2(1)*q1(2) + q2(2)*q1(1) - q2(3)*q1(4) + q2(4)*q1(3);
    n2 = q2(1)*q1(3) + q2(2)*q1(4) + q2(3)*q1(1) - q2(4)*q1(2);
    n3 = q2(1)*q1(4) - q2(2)*q1(3) + q2(3)*q1(2) + q2(4)*q1(1);

    q1q2 = [n0; n1; n2; n3];

    %% Rearrange result in the same dimensions as the two inputs
    [~,qCol] = size(q1);
    [~,rCol] = size(q2);
    if qCol == 4 && rCol == 4
        q1q2 = transpose(q1q2);
    end
end
```

Listing A.32: quatProdFcn2

```
function qp = quatProdFcn2(q1,q2)
%QUATPRODFCN2 Calculates the product of two quaternions. It takes two unit
% quaternions as input arguments and returns their product.
%
%  Input arguments: q1 and q2
%     The quaternions must be four-length row or column vectors or in a mixed
%     format (e.g., one could be a column vector and the other a row vector.)
%
%  Output argument: q1q2 = q1*q2
%     The output quaternion is also a four-length vector. It inherits its size
%     dimensions from the input argument, provided both inputs have similar sizes.
%     Otherwise, the output will be returned as a 4X1 column vector.
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

    %% check first the sizes of the the input arguments
```

```
    if length(q1) ~= 4 || length(q2) ~= 4
        error('The entered quaternions must be four-length vectors.')
    end

    %% calculate elements of the product of the two quaternions
    q1Re = q1(1);
    q1Im = q1(2:end);
    q2Re = q2(1);
    q2Im = q2(2:end);

    qpRe = q2Re*q1Re - dot(q2Im,q1Im);
    qpIm = q1Re*q2Im + q2Re*q1Im + cross(q1Im,q2Im);

    qp = [qpRe;qpIm];

    %% Rearrange result in the same dimensions as the two inputs (if they are the same)
    [~,qCol] = size(q1);
    [~,rCol] = size(q2);
    if qCol == 4 && rCol == 4
        qp = transpose(qp);
    end
end
```

Listing A.33: solverFcn

```
function [simulationSummary, param] = solverFcn(varargin)
%SOLVERFCN Automatically formulates the problem as a multi-body system and solves
% it numerically. If the total simulation time is longer than the maximum segment
% length defined in the dashboard file, this function solves the simulation in
% multiple segments and saves the solution results in a folder.
%
%  Input arguments
%     Input 1: A data structure containing all information needed to
%     formulate and solve the problem
%
%  Output argument
%     Output 1: A data structure containing summary parameters.
%     Output 2: The original parameters data structure updated with
% additional information generated during the simulation
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

tic
param = varargin{1};
dim = 7;                                % 3 position + 4 quaternions
stuckBeadLimit = param.solver.stuckBeadLimit;
numOfActive = 0;                        % no bead in the system at the beginning
tableTouchedRegister = [];              % the list of touched beads is empty
numOfFrozen = 0;                        % no bead is frozen at the beginning
segSol = struct;
simulationSummary = struct;
```

```
avfiberChar = struct;
tSpanOfEjectionInterval = [];
solOfEjectionInterval = [];
ejectionID = 0;
segmentID = 0;
simulationTime = param.sessionInfo.simulationTimeLength;
completedSimulationTime = 0;
remainingSimulationTime = simulationTime;
numericalTimeStep = param.solver.timeStep;
% numerical integration time step
numericalTimeStep_entire = transpose(0:numericalTimeStep:simulationTime);
ejectionIntervalTime = param.solver.ejectionTimeStep;
% one bead comes out of the nozzle per ejectionTimeStep
segmentIntervalTime = param.sessionInfo.simulationSegmentTimeLimit;
% total length of time the simulation will be run for
ejectionTimespanVectorLength = param.solver.numOfStepsPerEjection + 1;
% e.g., 0--10, 10 divisions but 11 time markers. Hence markers = divisions + 1;
numOfEjectionsInSimulation = ceil(simulationTime/ejectionIntervalTime);
numOfSegments = ceil(simulationTime/segmentIntervalTime);
segArraySizes = NaN(numOfSegments, 2);
fiberCharTable = NaN(numOfSegments, 4);
ti = 0;
tf = ti + ejectionIntervalTime;
globalNumOfRows = ceil(simulationTime/numericalTimeStep) + 100;
% with extra; it will be trimmed off at the end.
globalNumOfColumns = numOfEjectionsInSimulation + 10;
disp(['  Solution matix size (each):      ' ...
   num2str(globalNumOfRows) ' x ' num2str(globalNumOfColumns)]);
numOfRowsInSegment = ceil(segmentIntervalTime/numericalTimeStep) + 10;
% Solution holder matrix with extra rows
T_seg = NaN(numOfRowsInSegment,1);                        % Solution time vector
X_seg = NaN(numOfRowsInSegment,globalNumOfColumns);       % Solution x coordinates
Y_seg = NaN(numOfRowsInSegment,globalNumOfColumns);       % Solution y coordinates
Z_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Q1_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Q2_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Q3_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Q4_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
X_d_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Y_d_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
Z_d_seg = NaN(numOfRowsInSegment,globalNumOfColumns);
[nozP_entrie, nozV_entire, nozA_entire] = ...
   nozzlePosVelAccFcn(numericalTimeStep_entire, param.gCode);
param.gCode.nozTPVA = ...
   [numericalTimeStep_entire, nozP_entrie.', nozV_entire.', nozA_entire.'];
while completedSimulationTime < simulationTime
   segmentID = segmentID + 1;
   rowEndIndex = 1;
   if remainingSimulationTime < segmentIntervalTime
      segmentIntervalTime = remainingSimulationTime;
   end
   tf_segment = completedSimulationTime + segmentIntervalTime;
   while ti <= tf_segment
      ejectionID = ejectionID + 1;
```

```
83        disp(' ======================== SIMULATION STATUS ========================')
84        disp(param.sessionInfo.simulationDate);
85        disp(['Simulaiton name: ' param.sessionInfo.name]);
86        disp(['Run#:            ' num2str(param.solver.DOErunNumber)]);
87        disp(' ')
88        disp(' ---------------------------------------------------------------------')
89        localElapsedTime = toc;
90        Duration = duration(seconds(localElapsedTime));
91        Duration.Format = 'hh:mm:ss';
92        percentage = (ti/simulationTime)*100;
93        disp(['  Solver progress:                ' num2str(percentage,'%.0f') ...
94           ' % completed']);
95        disp('');
96        disp(['  Numerical time step:            ' num2str(param.solver.timeStep,...
97           '%.5f') ' s']);
98        disp(['  Virtual current simulation time: ' num2str(ti,'%.2f') ' s of ' ...
99           num2str(simulationTime,'%.2f') ' s']);
100       disp(['  Actual elapsed time:            ' char(Duration)]);
101       disp(['  Simulation segment length:      ' ...
102          num2str(param.sessionInfo.simulationTimeLength,'%.2f') ' s']);
103       disp('');
104       disp(['  Current simulation segment:     ' num2str(segmentID) ' of ' ...
105          num2str(numOfSegments)]);
106       disp(['  Number of beads ejected so far:  ' num2str(ejectionID)]);
107       disp(['  Solution matix size (each):     ' ...
108          num2str(globalNumOfRows) ' x ' num2str(globalNumOfColumns)]);
109       disp('');
110       disp(['  Feedrate:                       ' ...
111          num2str(param.gCode.feedRateOveride) ' mm/min ']);
112       %%  Sover
113       if isempty(solOfEjectionInterval) % the case at the start of new simulation
114          lastSol = [];
115       else
116          lastSol = solOfEjectionInterval(end,:);
117       end
118       %updateNumOfFrozenFn()
119       [lastActiveIndx, frozenThisPeriod, numOfFrozen] = ...
120          updateNumOfFrozenFcn(tableTouchedRegister, numOfActive, numOfFrozen, ...
121          stuckBeadLimit);
122       x0 = initialConditionFcn(lastSol,ti,param, numOfActive, ...
123          tableTouchedRegister, lastActiveIndx, dim);
124       tSpanOfEjectionIntervalExtra = ...
125          transpose(ti:numericalTimeStep:(tf + 3*numericalTimeStep));
126       solOfEjectionIntervalExtra = ...
127          zeros(length(tSpanOfEjectionIntervalExtra),length(x0));
128       numOfActive = ejectionID - numOfFrozen;
129       % the number of beads active during this iteration
130       tableTouchedRegister = [0,tableTouchedRegister(1:end-frozenThisPeriod)];
131       startingTimeIndx = 1;
132       while ti<tf
133          % if ti2 <  tf, contraint the stuck beads only and continue the simulation
134          % without adding more beads
135          tSpan_sub = ti:numericalTimeStep:(tf+3*numericalTimeStep);
136          % the time span definition for the remainder of the iteration
```

```
137         eqn = @(t, x)EOMFn(MultibodyClass(t,x,param, numOfActive, ...
138            tableTouchedRegister, x0, dim));
139         odeEvent = @(t,x)stickingEventFcn(t,x,param, tableTouchedRegister, ...
140            numOfActive, dim);
141         opts = odeset('RelTol',1e-6,'AbsTol',1e-6,'Events', odeEvent);
142         [timeSol_sub, sol_sub, ~, ~, ie] =  ode15s(eqn,tSpan_sub,x0, opts);
143         if any(ie) && timeSol_sub(end) <= tf && timeSol_sub(end) <= tf_segment
144            % check if another event was triggered during the sub-iteration
145            tableTouchedRegister(ie) = 1;
146            timeSol_sub(end) = timeSol_sub(end-1) + numericalTimeStep;
147            % clean up the last time step to give it equal spacing
148         end
149         endingTimeIndx = startingTimeIndx + length(timeSol_sub) - 1;
150         solOfEjectionIntervalExtra(startingTimeIndx:endingTimeIndx,:) = sol_sub;
151         ti = timeSol_sub(end);
152         x0 = sol_sub(end,:);
153         startingTimeIndx = endingTimeIndx;
154      end
155      tSpanOfEjectionInterval = ...
156         tSpanOfEjectionIntervalExtra(1:ejectionTimespanVectorLength,:);
157      % ... because of the tf+3*timeStep extra steps added to the timeOfPeriodExtra
158      solOfEjectionInterval = ...
159         solOfEjectionIntervalExtra(1:ejectionTimespanVectorLength,:);
160      ti = tSpanOfEjectionInterval(end);
161      tf = ti + ejectionIntervalTime;
162      %% Reposit results
163      X_ejectionIntervalSol = solOfEjectionInterval(:,1:dim:numOfActive*dim);
164      Y_ejectionIntervalSol = solOfEjectionInterval(:,2:dim:numOfActive*dim);
165      Z_ejectionIntervalSol = solOfEjectionInterval(:,3:dim:numOfActive*dim);
166      Q1_ejectionIntervalSol = solOfEjectionInterval(:,4:dim:numOfActive*dim);
167      Q2_ejectionIntervalSol = solOfEjectionInterval(:,5:dim:numOfActive*dim);
168      Q3_ejectionIntervalSol = solOfEjectionInterval(:,6:dim:numOfActive*dim);
169      Q4_ejectionIntervalSol = solOfEjectionInterval(:,7:dim:numOfActive*dim);
170      X_d_ejectionIntervalSol = ...
171         solOfEjectionInterval(:,numOfActive*dim+1:dim:2*numOfActive*dim);
172      Y_d_ejectionIntervalSol = ...
173         solOfEjectionInterval(:,numOfActive*dim+2:dim:2*numOfActive*dim);
174      Z_d_ejectionIntervalSol = ...
175         solOfEjectionInterval(:,numOfActive*dim+3:dim:2*numOfActive*dim);
176      if ~isequal(size(X_ejectionIntervalSol), size(Y_ejectionIntervalSol), ...
177            size(Z_ejectionIntervalSol), size(Q1_ejectionIntervalSol), ...
178            size(Q2_ejectionIntervalSol), size(Q3_ejectionIntervalSol), ...
179            size(Q4_ejectionIntervalSol), size(X_d_ejectionIntervalSol), ...
180            size(Y_d_ejectionIntervalSol), size(Z_d_ejectionIntervalSol))
181         error('Ejection interval solution matrices are not of euqal dimensions!')
182      end
183      rowStartIndex = rowEndIndex;
184      rowEndIndex = rowStartIndex + ejectionTimespanVectorLength - 1;
185      columnStartIndex = numOfFrozen + 1;
186      columnEndIndex = columnStartIndex + size(X_ejectionIntervalSol,2) - 1;
187      T_seg(rowStartIndex:rowEndIndex,:) = tSpanOfEjectionInterval;
188      X_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
189         flip(X_ejectionIntervalSol,2);
190      Y_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
```

```
        flip(Y_ejectionIntervalSol,2);
    Z_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Z_ejectionIntervalSol,2);
    Q1_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Q1_ejectionIntervalSol,2);
    Q2_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Q2_ejectionIntervalSol,2);
    Q3_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Q3_ejectionIntervalSol,2);
    Q4_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Q4_ejectionIntervalSol,2);
    X_d_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(X_d_ejectionIntervalSol,2);
    Y_d_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Y_d_ejectionIntervalSol,2);
    Z_d_seg(rowStartIndex:rowEndIndex,columnStartIndex:columnEndIndex) = ...
        flip(Z_d_ejectionIntervalSol,2);
    clc
end
% A segment simulation just ended here ....
trimEjectinIntervalIndex = find(tSpanOfEjectionInterval > tf_segment,1);
tSpanOfEjectionInterval(trimEjectinIntervalIndex:end,:) = [];
solOfEjectionInterval(trimEjectinIntervalIndex:end,:) = [];
ti = tSpanOfEjectionInterval(end);
tf = ti + ejectionIntervalTime;
completedSimulationTime = tf_segment;
remainingSimulationTime = simulationTime - tf_segment;
trimSegmentIndex = find(T_seg > completedSimulationTime,1);
if (trimSegmentIndex == 1) || isempty(trimSegmentIndex)
    error('There is something worong here!')
end
T_segTrimmed = T_seg(1:trimSegmentIndex-1,:);
X_segTrimmed = X_seg(1:trimSegmentIndex-1,:);
Y_segTrimmed = Y_seg(1:trimSegmentIndex-1,:);
Z_segTrimmed = Z_seg(1:trimSegmentIndex-1,:);
Q1_segTrimmed = Q1_seg(1:trimSegmentIndex-1,:);
Q2_segTrimmed = Q2_seg(1:trimSegmentIndex-1,:);
Q3_segTrimmed = Q3_seg(1:trimSegmentIndex-1,:);
Q4_segTrimmed = Q4_seg(1:trimSegmentIndex-1,:);
X_d_segTrimmed = X_d_seg(1:trimSegmentIndex-1,:);
Y_d_segTrimmed = Y_d_seg(1:trimSegmentIndex-1,:);
Z_d_segTrimmed = Z_d_seg(1:trimSegmentIndex-1,:);
[x_NaNCheck, ~] = lineNaNCheckerFcn(X_segTrimmed);
if x_NaNCheck == 1
    error('There is a problem here. NaN row found in solution!')
end
segSize = size(X_segTrimmed);
[nozzlePos_segmentTrimmed_verticalVectors, ~,~] = ...
    nozzlePosVelAccFcn(T_segTrimmed, param.gCode);
nozzlePos_segmentTrimmed = nozzlePos_segmentTrimmed_verticalVectors';
[X_MAT,Y_MAT,Z_MAT] = ...
    positionsFillerFcn(X_segTrimmed,Y_segTrimmed,Z_segTrimmed, ...
    nozzlePos_segmentTrimmed);
segSol.time_vec = T_segTrimmed;
```

```
245     segSol.x = X_MAT;
246     segSol.y = Y_MAT;
247     segSol.z = Z_MAT;
248     segSol.q1 = Q1_segTrimmed;
249     segSol.q2 = Q2_segTrimmed;
250     segSol.q3 = Q3_segTrimmed;
251     segSol.q4 = Q4_segTrimmed;
252     segSol.x_dot = X_d_segTrimmed;
253     segSol.y_dot = Y_d_segTrimmed;
254     segSol.z_dot = Z_d_segTrimmed;
255     segSol.nozzlePos.x = nozzlePos_segmentTrimmed(:,1);
256     segSol.nozzlePos.y = nozzlePos_segmentTrimmed(:,2);
257     segSol.nozzlePos.z = nozzlePos_segmentTrimmed(:,3);
258     segSol.param = param;
259     segSol.param.solver.segmentID = segmentID;
260     segSol.fiberChar = fiberCharactersticsFcn(segSol);
261     % updated with summary of fiber charactersitcs
262     segArraySizes(segmentID,:) = segSize;
263     fiberCharTable(segmentID,:) = [segSol.fiberChar.averageFiberDiameter, ...
264        segSol.fiberChar.maximumCollectionDiameter, ...
265        segSol.fiberChar.max_beadSpeed, segSol.fiberChar.percentage_elongation];
266     if lower(param.sessionInfo.inpectSegmentResults) == yes
267        plotForInspectionFn2(segSol, segmentID, param);
268     end
269     save([param.solver.solutionPath '/SegmentSoln' ...
270        num2str(segmentID) '.mat'],'segSol','-v7.3');
271     clear segSol
272     T_seg(~isnan(T_seg)) = NaN;
273     X_seg(~isnan(X_seg)) = NaN;
274     Y_seg(~isnan(Y_seg)) = NaN;
275     Z_seg(~isnan(Z_seg)) = NaN;
276     Q1_seg(~isnan(Q1_seg)) = NaN;
277     Q2_seg(~isnan(Q2_seg)) = NaN;
278     Q3_seg(~isnan(Q3_seg)) = NaN;
279     Q4_seg(~isnan(Q4_seg)) = NaN;
280     X_d_seg(~isnan(X_d_seg)) = NaN;
281     Y_d_seg(~isnan(Y_d_seg)) = NaN;
282     Z_d_seg(~isnan(Z_d_seg)) = NaN;
283     nonNaNlasetRowIndx = find(~isnan(X_segTrimmed(end,:)),1);
284     X_seg(1,1:nonNaNlasetRowIndx) = X_MAT(end,1:nonNaNlasetRowIndx);
285     Y_seg(1,1:nonNaNlasetRowIndx) = Y_MAT(end,1:nonNaNlasetRowIndx);
286     Z_seg(1,1:nonNaNlasetRowIndx) = Z_MAT(end,1:nonNaNlasetRowIndx);
287 end
288 weightCol = segArraySizes(:,1);
289 weightedFiberChar = weightCol.*fiberCharTable;
290 summedFiberChar = sum(weightedFiberChar,1);
291 averagedFiberChar = summedFiberChar/sum(weightCol,all);
292 avfiberChar.averageFiberDiameter = averagedFiberChar(1,1);
293 avfiberChar.maximumCollectionDiameter = averagedFiberChar(1,2);
294 avfiberChar.max_beadSpeed = averagedFiberChar(1,3);
295 avfiberChar.percentage_elongation = averagedFiberChar(1,4);
296 param.solver.numOfSegments = segmentID;
297 param.solver.segSolArraySizes = segArraySizes;
298 simulationSummary.param = param;
```

```
simulationSummary.fiberChar = avfiberChar;
save([param.solver.solutionPath ...
   '/SimulationSummary.mat'],'simulationSummary','-v7.3');
end
```

Listing A.34: stickingEventFcn

```
function [values,isterminal,direction] = stickingEventFcn(t,statesVec, param, ...
   tableTouchedRegister, numOfActive, dim)
%STICKINGEVENTFCN Checkes if any active beads stick to the collector
% plate. The collector distance is used as an event trigger criterion.
%
%  Input arguments
%     Input 1: current simulation time
%     Input 2: Vector of numerical integration variables
%     Input 3: Data structure containing parameter values
%     Input 4: A list of bead indexes touching the table
%     Input 5: Number of active beads in the current iteration
%     Input 6: Number of independent generalized coordinates per bead
%
%  Output argument
%     Output 1: Existing status of the beads (vector)
%     Output 2: List of newly touching beads (vector)
%     Output 3: Direction of approaching the plate (vector)
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%
   beadVectorProjection = zeros(numOfActive,1);
   [nozPos, ~, ~] = nozzlePosVelAccFcn(t, param.gCode);
   for i = 1:numOfActive
      idx1 = i*dim - 6;
      idx2 = idx1 + 2;

      beadVector = statesVec(idx1:idx2) - nozPos;
      beadVectorProjection(i,1) = dot(beadVector,param.equipment.nozzleUnitVec);
   end
   values = param.equipment.collectorDistance - beadVectorProjection;

   isterminal = ones(numOfActive,1) - tableTouchedRegister';
   direction = -1*ones(numOfActive, 1);
end
```

Listing A.35: updateNumOfFrozenFcn

```
function [lastActiveIndx, frozenThisPeriod, numOfFrozen] = ...
updateNumOfFrozenFcn(tableTouchedRegister,numOfActive,numOfFrozen,stuckBeadLimit)
%UPDATENUMBEROFFROZENFCN Keeps track of beads that are permanently stuck on the
%collector plate.
%
%  Input parameters:
%     Input 1: Exising list of beads' table-sticking status
%     Input 2: Number of active beads in the current iteration
%     Input 3: Total number of frozen beads so far
```

```
%     Input 4: The number of beads that are currently touching and allowed
%     to participate in the active simulation
%
%  Output parameters:
%     Output 1: The index of the latest last active bead after removing stuck beads
%               from the simulation
%     Output 2: Number of beads removed from the simulation during this iteration
%     Output 3: Updated number of frozen beads
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

if any(tableTouchedRegister)
    lastActiveIndx = find(tableTouchedRegister,1) + stuckBeadLimit - 1;
    % find the index of the first stuck bead.
    % Add stuckBeadLimit to it to find the last stuck bead index.
    if lastActiveIndx > numOfActive
        lastActiveIndx = numOfActive;
    end
    frozenThisPeriod = numOfActive - lastActiveIndx;
    % Determine if there are any beads frozen during this iteration (period).
    % 0 and negative values indicate no stuck beads.
else
    frozenThisPeriod = 0;
    lastActiveIndx = numOfActive;
end
numOfFrozen = numOfFrozen + frozenThisPeriod;
end
```

Listing A.36: vectorAngleFcn

```
function angleFromXAxis = vectorAngleFcn(targetVector, radOrDeg)
%
%  Author: Abiy Wubneh
%  Copyright 2023 MFCL@Mechanical Engineering, University of Alberta
%

xAxisVector = [1,0,0];
angleFromXAxis = atan2d(norm(cross(xAxisVector, targetVector)), ...
   dot(xAxisVector,targetVector));
% check if the target vector is in the 3rd or the 4th quadrant, ...
% i.e., is y value negative?
if targetVector(2) < 0
    angleFromXAxis = 360 - angleFromXAxis;
end
if lower(radOrDeg) == r
    angleFromXAxis = deg2rad(angleFromXAxis);
elseif lower(radOrDeg) ~= d
    error(['Unknown angle format passed as a second argument! Please provide as ' ...
       'a second argment either r for radians or d for degree'])
end
end
```

Appendix B: DOE Data

Table B.1

Std. Order	*Run Order*	Flow rate ($mL\,h^{-1}$)	Voltage (kV)	Distance (mm)	Temp (°C)	Collection diameter (mm)	Average fiber diameter (µm)
33	*1*	2	15	40	250	22.04	58.09
56	*2*	3	15	30	240	20.63	99.38
55	*3*	3	15	30	230	18.81	133.49
14	*4*	1	17.5	40	240	16.33	67.05
48	*5*	2	20	30	250	19.97	54.83
72	*6*	3	17.5	50	250	24.85	62.47
27	*7*	1	20	50	250	19.45	39.57
79	*8*	3	20	50	230	22.50	85.97
67	*9*	3	17.5	40	230	20.46	120.98
38	*10*	2	17.5	30	240	17.44	78.44
63	*11*	3	15	50	250	25.53	76.71
39	*12*	2	17.5	30	250	18.19	67.12
69	*13*	3	17.5	40	250	23.30	68.19
71	*14*	3	17.5	50	240	22.01	88.62
75	*15*	3	20	30	250	19.44	70.66
16	*16*	1	17.5	50	230	17.47	66.24
62	*17*	3	15	50	240	24.24	103.39
29	*18*	2	15	30	240	16.37	103.95
73	*19*	3	20	30	230	18.55	120.50
8	*20*	1	15	50	240	18.20	79.49
42	*21*	2	17.5	40	250	21.96	54.93
12	*22*	1	17.5	30	250	17.31	51.82
9	*23*	1	15	50	250	21.20	43.13
46	*24*	2	20	30	230	16.87	81.15
64	*25*	3	17.5	30	230	18.05	135.86
20	*26*	1	20	30	240	13.96	79.05
28	*27*	2	15	30	230	16.44	125.72
22	*28*	1	20	40	230	13.54	91.84
57	*29*	3	15	30	250	19.18	84.87
3	*30*	1	15	30	250	15.52	58.16
78	*31*	3	20	40	250	23.53	68.68
52	*32*	2	20	50	230	20.60	85.02

Table B.1 continued from previous page

Std. Order	*Run Order*	Flow rate ($mL\,h^{-1}$)	Voltage (kV)	Distance (mm)	Temp (°C)	Collection diameter (mm)	Average fiber diameter (µm)
15	*33*	1	17.5	40	250	18.01	62.53
58	*34*	3	15	40	230	21.72	100.31
44	*35*	2	17.5	50	240	20.27	79.69
10	*36*	1	17.5	30	230	14.65	88.79
18	*37*	1	17.5	50	250	20.37	48.27
6	*38*	1	15	40	250	19.07	45.92
21	*39*	1	20	30	250	15.37	75.96
54	*40*	2	20	50	250	22.67	43.99
26	*41*	1	20	50	240	28.88	42.90
4	*42*	1	15	40	230	18.28	63.51
23	*43*	1	20	40	240	19.15	41.04
13	*44*	1	17.5	40	230	15.68	57.53
80	*45*	3	20	50	240	25.05	78.33
35	*46*	2	15	50	240	20.76	84.69
53	*47*	2	20	50	240	20.93	61.17
2	*48*	1	15	30	240	15.34	59.40
41	*49*	2	17.5	40	240	19.44	64.70
1	*50*	1	15	30	230	14.45	74.56
31	*51*	2	15	40	230	18.86	106.04
32	*52*	2	15	40	240	19.90	88.14
40	*53*	2	17.5	40	230	18.16	94.05
81	*54*	3	20	50	250	23.50	65.62
76	*55*	3	20	40	230	20.74	101.31
30	*56*	2	15	30	250	17.01	84.02
59	*57*	3	15	40	240	22.40	93.09
61	*58*	3	15	50	230	22.13	119.57
77	*59*	3	20	40	240	20.04	112.26
24	*60*	1	20	40	250	18.55	39.87
25	*61*	1	20	50	230	19.44	50.00
37	*62*	2	17.5	30	230	18.40	89.97
34	*63*	2	15	50	230	20.58	104.17
68	*64*	3	17.5	40	240	21.67	101.28
47	*65*	2	20	30	240	17.69	69.13
51	*66*	2	20	40	250	20.84	46.69
60	*67*	3	15	40	250	23.63	78.64
11	*68*	1	17.5	30	240	15.04	57.33
17	*69*	1	17.5	50	240	19.96	50.09
49	*70*	2	20	40	230	18.87	71.95
19	*71*	1	20	30	230	14.59	61.58
45	*72*	2	17.5	50	250	23.48	56.49
50	*73*	2	20	40	240	20.17	58.24
7	*74*	1	15	50	230	17.52	75.77
74	*75*	3	20	30	240	19.06	93.30
43	*76*	2	17.5	50	230	19.63	88.70

Table B.1 continued from previous page

Std. Order	*Run Order*	Flow rate ($mL\,h^{-1}$)	Voltage (kV)	Distance (mm)	Temp (°C)	Collection diameter (mm)	Average fiber diameter (µm)
66	*77*	3	17.5	30	250	20.01	77.44
36	*78*	2	15	50	250	22.91	63.64
5	*79*	1	15	40	240	17.30	49.00
70	*80*	3	17.5	50	230	21.47	113.06
65	*81*	3	17.5	30	240	19.04	97.27

Table B.2: Statistical summary of DOE data

	μ	σ	**95% CI (μ)**	**95% CI (σ)**	$p-value$	ks	cv
Collection diameter	2.9	2.4	[2.4, 3.5]	[2.1, 2.8]	0.0011	0.1351	0.1163
Average fiber diameter	8.5	7.2	[6.9, 10.2]	[6.2, 8.6]	0.0035	0.1259	0.1163