

Position-Aware Control of Myoelectric Prostheses

by

Heather E. Williams

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

Doctor of Philosophy

Biomedical Engineering  
University of Alberta

© Heather E. Williams, 2024

# Abstract

As reported in 2020, millions of individuals worldwide have some form of *upper limb* amputation or congenital loss of limb that impedes execution of everyday actions like reaching for, grasping, transporting, and releasing objects. Although lost function can be artificially replaced via prostheses, there remains a problem—state-of-the-art robotic (myoelectric) devices continue to cause frustrating control challenges for users.

The work herein focuses on mitigating the sources of myoelectric prosthesis control problems for those with below elbow (transradial) amputation. Wrist and hand movements of such devices are controlled by captured electromyographic (EMG) signals that are generated by muscles in a user’s residual limb. Advanced pattern recognition-based controllers now make device operation intuitive for users, with software models trained to recognize EMG signal patterns. Still, devices can become unreliable when limb position changes are introduced during device use. This specific problem is known as the limb position effect. Here, when a user attempts to use their prosthesis in a position not included in model training, the captured EMG signals are unrecognizable and misinterpreted by the controller. This scenario can cause control malfunctions that result in unexpected device movements. Understandably, unreliable control leads to user frustration.

The *broad goal* of this thesis, therefore, is to reimagine upper limb myoelectric prosthesis control—offering persons with transradial amputation reliable, robust, and personalized device operation. Its *specific objective* is to develop an improved prosthesis control solution that mitigates the limb position effect.

The work in this thesis addresses the limb position effect problem by implementing *position-aware* myoelectric prosthesis control. For such control, EMG data must be augmented with details about

a user's limb position during device use. This was accomplished by capturing positional data via an inertial measurement unit (IMU) device worn on a user's forearm, with blended EMG and IMU data handled using a deep neural network. Several complex recurrent convolutional neural network (RCNN) models for pattern recognition-based control were explored in this thesis, as they offer two major advantages: (1) they can be trained using large multimodal datasets (including EMG and IMU data), and (2) their architectures can learn features directly from input data to yield accurate movement predictions. After assessing the accuracy of both classification- and regression-based RCNN models, the three most promising were compared to a commonly implemented baseline classification model. Comparative testing sessions required participants *without* upper limb impairment to perform functional tasks in multiple limb positions. They wore a simulated prosthesis (controlled by each model), as a proxy for device use by individuals with limb loss. An RCNN regression-based model yielded the best task performance versus the baseline alternative. Given this, a participant *with* amputation was recruited to assess the novel control solution, confirming that it could indeed provide control across multiple limb positions that is accurate, simultaneous, and proportional to EMG signal intensity.

This work contributes, for the first time, a ***position-aware deep learning regression-based myoelectric control solution*** for use by persons with transradial amputation. The solution is an exciting new direction for prosthesis control—a departure from commonly used classification approaches. After future refinements to and testing of this solution, a long-term goal is to see it incorporated into prosthesis designs of today and tomorrow. Future extensions to this work could see regression-based control implemented in context-specific human movement therapies and technologies. Ultimately, the benefits of reliable control are far-reaching.

# Preface

This thesis is an original work by Heather E. Williams. The research project, of which this thesis is a part of, received research ethics approval from the University of Alberta Health Research Ethics Board (Pro00086557).

This thesis contains information published in seven manuscripts currently published or submitted.

The majority of Chapter 2 has been published as:

Ahmed W. Shehata, **Heather E. Williams**, Jacqueline S. Hebert, Patrick M. Pilarski, “Machine learning for the control of prosthetic arms: using electromyographic signals for improved performance,” *IEEE Signal Processing Magazine* 38(4):46–53, June 2021.

The first author role was shared with A. W. Shehata. Contributions as co-first author: figure generation; writing and revisions of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews. I was the only student author on this manuscript.

The majority of Chapter 3 has been published as:

**Heather E. Williams**, Ahmed W. Shehata, Michael R. Dawson, Erik Scheme, Jacqueline S. Hebert, Patrick M. Pilarski, “Recurrent Convolutional Neural Networks as an Approach to Position-Aware Myoelectric Prosthesis Control,” *IEEE Transactions on Biomedical Engineering* 69(7):2243–55, July 2022.

Contributions as first author: experimental protocol design; human participant scheduling; collection, processing, and analysis of experimental data; development of machine learning models; figure and table generation; writing and revisions of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

The majority of Chapter 4 has been published as:

**Heather E. Williams**, Johannes Günther, Jacqueline S. Hebert, Patrick M. Pilarski, Ahmed W. Shehata, “Composite Recurrent Convolutional Neural Networks Offer a Position-Aware Prosthesis Control Alternative While Balancing Predictive Accuracy with Training

Burden,” *Proceedings of the International Conference for Rehabilitation Robotics (ICORR)*, July 25–29, 2022, Rotterdam, the Netherlands.

Contributions as first author: experimental protocol design; processing and analysis of experimental data; development of machine learning models; figure generation; writing and revisions of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

The majority of Chapter 5 has been published as:

**Heather E. Williams**, Jacqueline S. Hebert, Patrick M. Pilarski, Ahmed W. Shehata, “A Case Series in Position-Aware Myoelectric Prosthesis Control using Recurrent Convolutional Neural Network Classification with Transfer Learning,” *Proceedings of the International Conference for Rehabilitation Robotics (ICORR)*, September 24–28, 2023, Singapore. (accepted for podium presentation)

Contributions as first author: experimental protocol design; fabrication modifications to experimental apparatuses; human participant scheduling; collection, processing, and analysis of experimental data; development of machine learning models; figure generation; writing and revisions of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

The majority of Chapter 6 has been accepted as:

**Heather E. Williams**, Ahmed W. Shehata, Kodi Y. Cheng, Jacqueline S. Hebert, Patrick M. Pilarski, “A multifaceted suite of metrics for comparative myoelectric prosthesis controller research,” *PLOS One* 19(5): e0291279, May 2024.

Contributions as first author: experimental protocol design; fabrication modifications to experimental apparatuses; human participant scheduling; collection, processing, and analysis of experimental data; development of machine learning models; figure and table generation; writing and revisions of draft manuscripts for co-author review; and preparation of manuscript revisions in response to peer reviews.

The majority of Chapter 7 has been submitted as:

**Heather E. Williams**, Ahmed W. Shehata, Kodi Y. Cheng, Jacqueline S. Hebert, Patrick M. Pilarski, “Myoelectric Prosthesis Control using Recurrent Convolutional Neural Network Regression Mitigates the Limb Position Effect,” submitted to *Neural Computing and Applications*, February 2024.

Contributions as first author: experimental protocol design; fabrication modifications to experimental apparatuses; human participant scheduling; collection, processing, and analysis of experimental data; development of machine learning models; figure and table generation; and writing and revisions of draft manuscripts for co-author review.

The majority of Chapter 8 has been accepted as:

**Heather E. Williams**, Jacqueline S. Hebert, Patrick M. Pilarski, Ahmed W. Shehata, “Evidence that a deep learning regression-based controller mitigates the limb position effect for an individual with transradial amputation,” accepted to *Myoelectric Controls (MEC) Symposium*, March 2024.

Contributions as first author: experimental protocol design; fabrication modifications to experimental apparatuses; human participant scheduling; collection, processing, and analysis of experimental data; development of machine learning models; figure generation; and writing and revisions of draft manuscripts for co-author review.

# Dedication

To my Mom and Dad—I could not have done it without you!

# Acknowledgements

Thank you to my dream team of PhD supervisory committee: Dr. Jacqueline Hebert (who I think also goes by the name *Super Woman* and is the best mentor any female researcher could ever hope for), Dr. Patrick Pilarski (who brings joy to learning, research, and purple pen editing), and Dr. Albert Vette (who made my unforgettable Japanese summer research happen and who shaped my communication skills).

Additionally, thank you to the unwavering support of my BLINC lab family: Dr. Ahmed Shehata, for guiding much of my hands-on learning, my *what if...* investigations, and for being present for all of my data collection sessions; Kodi Cheng, for her dedication to data collection and GaMA data processing; Quinn Boser and Riley Dawson for help with all things GaMA-related—Quinn, for help with calculations, troubleshooting, and pitching in with data collection—Riley, for help with GaMA troubleshooting and for getting my data synchronization set up; and finally to current and former lab members for inspiring me to keep on learning, especially Rory Dawson, Katherine Schoepp, and Dr. Jon Schofield.

Of course, none of my work would have been possible without participants. To each and every one who gave of their time to help with my research, I am forever grateful. Thank you also to the researchers with whom I've had the pleasure of collaborating. In particular, Dr. Erik Scheme, Dr. Johannes Gunther, Dr. Matija Milosevic, and Dr. Taishin Nomura.

I appreciate that I have a great support system that I could lean on throughout my PhD journey. Thank you to BFFs Maegan, Rana, Sarah, and Emily. Thank you to Rory, Katrina, Dylan, and Alex for much needed Monday night movie club breaks. Thank you to Dr. Lorne Sawula (a.k.a. Dr. Volleyball) who remains my lifelong coach. Finally, thank you to my parents and my extended family—"little Heather" did it!

I would also like to acknowledge the funding that made my research possible. Thank you to the Natural Sciences and Engineering Research Council of Canada, Killam Trusts, Philanthropic Educational Organization (especially Chapter Q), Canadian Federation of University Women Edmonton Club, Alberta Innovates, the Faculty of Engineering, Faculty of Medicine and Dentistry,

the Faculty of Graduate & Postdoctoral Studies, the University of Alberta, and Amii for their support. I am truly grateful.

# Table of Contents

<b>Chapter 1. Introduction.....</b>	<b>1</b>
<b>1.1. Problem Definition .....</b>	<b>1</b>
<b>1.2. Research Objectives &amp; Contributions.....</b>	<b>4</b>
<b>1.3. Thesis Outline .....</b>	<b>4</b>
<b>Chapter 2. Machine Learning for the Control of Myoelectric Upper Limb Prostheses .....</b>	<b>7</b>
<b>Abstract.....</b>	<b>8</b>
<b>2.1. Introduction .....</b>	<b>8</b>
2.1.1. Conventional Control Strategies.....	9
2.1.2. Machine Learning Control Strategies.....	10
<b>2.2. EMG Signal Processing Overview .....</b>	<b>10</b>
2.2.1. Filtering .....	10
2.2.2. Feature Extraction .....	11
<b>2.3. Machine Learning in EMG Signal-Driven Prosthesis Control.....</b>	<b>11</b>
2.3.1. Classification Control Strategy.....	11
2.3.2. Regression Control Strategy.....	14
2.3.3. Hybrid Control Strategy .....	17
2.3.4. Switching Control Strategy .....	18
2.3.5. Movement-Based Control Strategy .....	19
<b>2.4. Conclusion and Future Direction.....</b>	<b>20</b>
<b>Chapter 3. Recurrent Convolutional Neural Networks (RCNNs) for Position-Aware Control of Myoelectric Upper Limb Prostheses .....</b>	<b>22</b>
<b>Abstract.....</b>	<b>23</b>
<b>3.1. Introduction .....</b>	<b>23</b>
<b>3.2. Methods .....</b>	<b>26</b>
3.2.1. Participants .....	26
3.2.2. Experimental Setup .....	27

3.2.3. Data Collection.....	27
3.2.4. Data Pre-Processing.....	28
3.2.5. RCNN and Comparative Statistical Models.....	29
3.2.5.1. RCNN Models – Architecture .....	29
3.2.5.2. Comparative Statistical Model .....	31
3.2.6. Classification and Regression.....	31
3.2.6.1. Limb Position Classification.....	32
3.2.6.2. Movement Classification .....	33
3.2.6.3. Movement Regression .....	33
3.2.7. Outcome Measures and Statistical Analysis.....	34
3.2.7.1. Prediction Accuracy Calculations.....	34
3.2.7.2. Confusion Matrices.....	36
3.2.7.3. Overall Comparisons .....	36
<b>3.3. Results.....</b>	<b>36</b>
3.3.1. Limb Position Classification .....	36
3.3.2. Position-Aware Movement Classification: S1–S3 .....	37
3.3.3. Position-Aware Movement Regression: S1–S3 .....	39
3.3.4. Results Summary.....	42
<b>3.4. Discussion .....</b>	<b>43</b>
3.4.1. Position-Aware Classification .....	44
3.4.2. Position-Aware Regression .....	44
3.4.3. Promising RCNN Outcomes .....	46
3.4.4. Limitations.....	47
3.4.5. Future Work.....	47
<b>3.5. Conclusion.....</b>	<b>48</b>
<b>Chapter 4. Composite RCNN Classification Control .....</b>	<b>49</b>
<b>Abstract.....</b>	<b>50</b>
<b>4.1. Introduction .....</b>	<b>50</b>
<b>4.2. Methods .....</b>	<b>52</b>
4.2.1. Participants .....	53
4.2.2. Experimental Setup .....	53
4.2.3. Data Collection.....	53
4.2.4. Data Pre-Processing.....	55
4.2.5. RCNN-Based Model Architecture .....	55

4.2.5.1. Model Hyperparameters .....	55
4.2.6. Model Training & Testing.....	57
4.2.6.1. Model Training Process.....	57
4.2.6.2. Model Testing Process.....	58
<b>4.3. Results.....</b>	<b>59</b>
4.3.1. Generalized Model Performance .....	59
4.3.2. Personalized model performance.....	59
4.3.3. Composite Model Performance.....	60
4.3.4. Results Summary.....	60
4.3.5. Overfitting assessment.....	61
<b>4.4. Discussion .....</b>	<b>61</b>
<b>4.5. Conclusion.....</b>	<b>63</b>
<b>Chapter 5. RCNN Classification Control with Transfer Learning.....</b>	<b>65</b>
<b>Abstract.....</b>	<b>66</b>
<b>5.1. Introduction .....</b>	<b>67</b>
<b>5.2. Methods .....</b>	<b>69</b>
5.2.1. Participants .....	69
5.2.2. EMG and Accelerometer Data Collection.....	70
5.2.3. Donned Simulated Prosthesis.....	70
5.2.4. Control Model Implementation and Training.....	71
5.2.5. Data Processing .....	73
5.2.6. Control Practice .....	74
5.2.7. Motion Capture Setup.....	74
5.2.8. Functional Tasks for Control Testing.....	74
5.2.9. Experimental Data Analysis .....	75
<b>5.3. Results.....</b>	<b>76</b>
5.3.1. Significant Differences Between Controllers.....	77
5.3.2. Task-Specific Observations Between Controllers.....	77
<b>5.4. Discussion .....</b>	<b>78</b>
5.4.1. TL Impact on Training Burden.....	78
5.4.2. Pre-Training Conditions .....	78
5.4.3. Comprehensive Metrics Needed.....	79
<b>5.5. Conclusion.....</b>	<b>80</b>

**Chapter 6. A Suite of Metrics for Comparative Myoelectric Prosthesis Control Research 81**

**Abstract .....82**

**6.1. Introduction .....82**

**6.2. Introducing the Suite of Control Evaluation Metrics .....85**

**6.3. Overview: Reinvestigation using our Suite of Metrics.....90**

**6.4. Methods .....91**

6.4.1. Participants ..... 92

6.4.2. Signal Data Collection & Processing Procedure ..... 93

6.4.3. Simulated Prosthesis & Donning Procedure ..... 94

6.4.4. Control Model Descriptions & Training Routines ..... 94

6.4.5. Simulated Device Control Practice & Testing Eligibility ..... 97

6.4.6. Motion Capture Setup & Kinematic Calibrations ..... 97

6.4.7. Functional Tasks & Data Collection ..... 99

6.4.8. Survey Administration..... 99

6.4.9. Data Processing & Calculation Procedures ..... 100

6.4.10. Statistical Analysis ..... 102

6.4.11. Limb Position Effect Identification ..... 103

**6.5. Results.....105**

6.5.1. Task Performance ..... 105

6.5.2. Control Characteristics ..... 105

6.5.3. User Experience..... 109

**6.6. Discussion .....109**

6.6.1. Findings from Control Characteristics Metrics ..... 110

6.6.2. Findings from Task Performance Metrics ..... 111

6.6.3. Findings from User Experience Metrics..... 113

6.6.4. Evidence of Training Routine Reduction ..... 113

6.6.5. Limitations ..... 114

6.6.6. Future Work..... 114

**6.7. Conclusion .....115**

**Chapter 7. A Comparison of RCNN Classification and Regression Control..... 116**

**Abstract .....117**

**7.1. Introduction .....117**

<b>7.2. Methods</b> .....	<b>121</b>
7.2.1. Overview .....	121
7.2.2. Participants .....	122
7.2.3. Muscle Signal Data Collection.....	123
7.2.3.1. Myo Gesture Control Armband .....	123
7.2.3.2. Simulated Prosthesis.....	124
7.2.3.3. EMG and Accelerometer Data Processing .....	124
7.2.4. Controller Descriptions: Model Architectures, Training Routines, & Implementation .....	125
7.2.4.1. RCNN-Class Controller.....	125
7.2.5. RCNN-Reg Controller.....	127
7.2.5.1. LDA-Baseline Controller.....	128
7.2.5.2. Control Model Implementation .....	128
7.2.6. Simulated Device Control Practice & Testing Eligibility .....	129
7.2.7. Motion Capture Setup & Kinematic Calibrations .....	130
7.2.8. Controller Testing.....	131
7.2.8.1. Functional Task Execution .....	131
7.2.8.2. Perceived Control via Participant Survey Administration.....	132
7.2.9. Analysis of Controller Testing Data.....	133
7.2.9.1. Motion Capture Data Processing.....	133
7.2.9.2. Finalization of Grip Aperture and Wrist Rotation Measures.....	133
7.2.9.3. Task Segmentation.....	133
7.2.9.4. Control Assessment Metrics Calculated.....	134
7.2.9.5. Statistical Analysis.....	137
<b>7.3. Results</b> .....	<b>137</b>
7.3.1. Significant Differences Between RCNN-Based Controllers and LDA-Baseline.....	138
7.3.1.1. Discrete Metric Example: Number of Grip Aperture Adjustments.....	139
7.3.1.2. Continuous Metric Example: Grip Aperture Plateaus .....	140
7.3.2. Metrics-Substantiated Evidence of Limb Position Effect .....	141
7.3.3. RCNN-Reg Simultaneous and Proportional Velocity Control Capabilities.....	141
<b>7.4. Discussion</b> .....	<b>141</b>
<b>7.5. Conclusion</b> .....	<b>145</b>
<b>Chapter 8. RCNN Regression Control: Tested by an Individual with Amputation</b> .....	<b>147</b>
<b>Abstract</b> .....	<b>148</b>
<b>8.1. Introduction</b> .....	<b>148</b>

<b>8.2. Methods .....</b>	<b>150</b>
<b>8.3. Results.....</b>	<b>151</b>
<b>8.4. Discussion &amp; Future Work.....</b>	<b>153</b>
<b>8.5. Conclusion.....</b>	<b>154</b>
<b>Chapter 9. Conclusions and Future Directions .....</b>	<b>155</b>
<b>References .....</b>	<b>158</b>
<b>Appendix A. Offline Analysis Prediction Accuracies.....</b>	<b>176</b>
<b>Appendix B. Offline Analysis Training and Prediction Times .....</b>	<b>178</b>
<b>Appendix C. Table of Metrics Examples.....</b>	<b>180</b>
<b>Appendix D. Detailed Controller Comparison Results.....</b>	<b>184</b>

# List of Figures

Figure 1-1. The inner workings of an example myoelectric prosthetic limb on an individual with transradial amputation.....	2
Figure 1-2. Summary of thesis chapters.....	6
Figure 2-1. Research question addressed by Chapter 2 and remaining gap.....	7
Figure 2-2. Overview of EMG signal-driven upper limb prosthesis control.....	9
Figure 2-3. Visualization of class decision boundaries between hand open and hand close using LDA or SVM.....	13
Figure 2-4. Visualization of regressor output over DOF 1 rotation and DOF 2 hand open/close using multivariate linear regression SVR.....	16
Figure 2-5. Hybrid control requiring 2 input signals to the prosthetic limb.....	18
Figure 2-6. Switching control requiring 2 input signals to the prosthetic limb.....	19
Figure 3-1. Research question addressed by Chapter 3 and remaining gaps.....	22
Figure 3-2. The placement of each Myo armband, along with limb and movements used in data collection.....	27
Figure 3-3. RCNN architecture.....	30
Figure 3-4. Active and inactive periods for each degree of freedom.....	35
Figure 3-5. Mean limb position classification accuracy .....	37
Figure 3-6. Mean movement classification accuracy.....	38
Figure 3-7. RCNN movement classification confusion matrices.....	40
Figure 3-8. Mean movement regression $R^2$ values.....	40
Figure 3-9. RCNN movement regression confusion matrices.....	42

Figure 4-1.	Research question addressed by Chapter 4.....	49
Figure 4-2.	The placement of each Myo armband, along with limb positions and movements used in data collection.....	54
Figure 4-3.	Generalized, Personalized, and Composite RCNN-based myoelectric prosthesis control models.....	56
Figure 4-4.	Mean predictive accuracy and cumulative training routine durations for the Generalized, Personalized, and Composite Models.....	59
Figure 4-5.	Mean predictive accuracy of the Personalized, Generalized, and Composite Models under time-delayed conditions.....	60
Figure 4-6.	Conceptual illustration of how movement predictive accuracy might expectedly vary over extended durations of use.....	62
Figure 5-1.	Research questions addressed by Chapter 5 and remaining gap.....	66
Figure 5-2.	Myo armband and donned simulated prosthesis on a participant's forearm.....	71
Figure 5-3.	Architecture of RCNN-TL's model.....	72
Figure 5-4.	Pasta Box Task and Refined Clothespin Relocation Test trial movement order and setup.....	75
Figure 5-5.	Task performance metrics results with RCNN-TL and LDA-Baseline during Pasta, RCRT Up, and RCRT Down tasks.....	77
Figure 6-1.	Research question addressed by Chapter 6 and remaining gap.....	81
Figure 6-2.	RCNN-TL and LDA-Baseline's model training and testing.....	92
Figure 6-3.	Myo armband and simulated prosthesis on a participant's forearm.....	93
Figure 6-4.	Architecture of RCNN-TL's model.....	95
Figure 6-5.	Motion capture markers affixed to the simulated prosthesis.....	98

Figure 6-6.	Task setup for Pasta and RCRT Up and Down.....	100
Figure 6-7.	Box plots indicating LDA-Baseline number of wrist rotation adjustments in each RCRT Down Release and RCRT Down Grasp of each task movement.....	103
Figure 6-8.	Box plots indicating user experience metrics results with RCNN-TL and LDA-Baseline for: NASA-TLX, and usability survey.....	109
Figure 7-1.	Research question addressed by Chapter 7 and remaining gap.....	116
Figure 7-2.	Overview of controller testing sessions: RCNN-Class Session, RCNN-Reg Session, and LDA-Baseline Session.....	121
Figure 7-3.	Myo armband and simulated prosthesis on a participant’s forearm, with motion capture markers affixed to the simulated prosthesis.....	123
Figure 7-4.	Architecture of RCNN-Class’s model and RCNN-Reg’s model.....	126
Figure 7-5.	Task setup for Pasta and RCRT Up and Down, including task dimensions.....	132
Figure 7-6.	Box plots indicating LDA-Baseline number of wrist rotation adjustments in RCRT Down Grasp and RCRT Down Transport of each task movement.....	136
Figure 7-7.	Number of significant differences between RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline .....	138
Figure 7-8.	Median Number of Grip Aperture Adjustments in RCRT Down Grasp phases...	139
Figure 7-9.	Median Grip Aperture Plateaus in RCRT Down Reach-Grasps.....	140
Figure 7-10.	Simultaneous and proportional velocity control using RCNN-Reg.....	142
Figure 8-1.	Research question addressed by Chapter 8.....	147
Figure 8-2.	Equipment, experimentation, and analysis metrics in order of use.....	151
Figure 8-3.	Controller comparison results.....	152
Figure 9-1.	Summary of thesis chapters, with gaps for future work presented.....	156

# List of Tables

Table 3-1.	Regression standard deviations in inactive periods.....	41
Table 3-2.	Movement prediction accuracy summary.....	43
Table 6-1.	Description of Task Performance metrics used in analysis.....	88
Table 6-2.	Description of Control Characteristics metrics used in analysis.....	88
Table 6-3.	Description of User Experience metrics used in analysis.....	90
Table 6-4.	Limb position effect identification rules for control characteristics metrics.....	104
Table 6-5.	Task Performance metrics results.....	107
Table 6-6.	Control Characteristics metrics results.....	107
Table 7-1.	Summary of metrics used in this study.....	135
Table A-1.	Mean limb position classification accuracies.....	176
Table A-2.	Mean movement classification accuracies.....	177
Table A-3.	Mean movement regression R <sup>2</sup> values.....	177
Table B-1.	Mean limb position classification training and prediction times.....	178
Table B-2.	Mean movement regression training and prediction times.....	179
Table B-3.	Mean movement classification training and prediction times.....	179
Table D-1.	Task Performance Results.....	185
Table D-2.	Control Characteristics Results.....	186
Table D-3.	User Experience Results.....	187

# List of Abbreviations

<b>Abbreviation</b>	<b>Meaning</b>	<b>Chapters in which abbreviation was used</b>
∴	Because	6
ANN	Artificial Neural Network	2
BC	Baseline Classifier	3
BR	Baseline Regressor	3
CNN	Convolutional Neural Network	3
DOF	Degree Of Freedom	2, 3
EMG	Electromyography / Electromyographic	1, 2, 3, 4, 5, 6, 7, 8, 9
IMU	Inertial Measurement Unit	1, 2, 3, 4, 5, 6, 7, 8, 9
IQR	Interquartile Range	5, 6
LDA	Linear Discriminant Analysis	2, 3, 4, 5, 6, 7
LDA-Baseline	Linear Discriminant Analysis-based Baseline classification controller	5, 6, 7, 8
LSTM	Long Short-Term Memory	3, 4, 5, 6, 7, 8
MAPP	Modular-Adaptable Prosthetic Platform	5, 6, 7
NASA-TLX	National Aeronautics and Space Administration Task Load Index	6, 7
Pasta	Pasta Box Task	5, 6, 7, 8
PROS-TLX	Prosthesis Task Load Index	6, 7
PS1–PS9	Pilot Study Trial 1–9	4
R <sup>2</sup>	Coefficient of determination	3
RCNN	Recurrent Convolutional Neural Network	1, 3, 4, 5, 6, 7, 8, 9
RCNN-Class	Recurrent Convolutional Neural Network-based Classification controller	7
RCNN-Reg	Recurrent Convolutional Neural Network-based Regression controller	7, 8
RCNN-TL	Recurrent Convolutional Neural Network-based classification controller with Transfer Learning	5, 6
RCRT	Refined Clothespin Relocation Test	5, 6, 7, 8
ReLU	Rectified Linear Unit	3, 4, 5, 6, 7, 8
RMANOVA	Repeated-Measures Analysis Of Variance	5, 6, 7
RMSE	Root Mean Square Error	3
RNN	Recurrent Neural Network	3

<b>Abbreviation</b>	<b>Meaning</b>	<b>Chapters in which abbreviation was used</b>
S1–S3	Specification 1–3	3
SP	Simulated Prosthesis	5, 6
SVM	Support Vector Machine	2
SVR	Support Vector Regression	2, 3
TL	Transfer Learning	5, 6

# Chapter 1.

## Introduction

### 1.1. Problem Definition

Activities of daily living often require motor skills such as reaching for, grasping, and manipulating objects. Unfortunately, millions of individuals worldwide have some form of *upper limb* amputation or congenital loss of limb that impedes execution of these skills, including loss of finger(s), hand, or arm [1], [2], [3]. For those with below elbow, or *transradial amputation*, prosthesis solutions must replace the functions of the wrist joint and hand. Replacing lost function with a robotic, or *myoelectric*, prosthetic limb, however, continues to pose control challenges for users, particularly as the device must reliably perform intricate hand movements in a range of positions around the body [4]. Although state-of-the-art prostheses are commercially available, unreliable control has been cited as the main reason for device abandonment [5].

Unreliable control is characterized by prosthesis actions that do not align with user intentions. Control of a myoelectric upper limb prosthesis is typically based on electromyographic (EMG) signals generated by the muscles of a user's residual limb, to indicate their device movement intention [6]. That is, when they contract a specific muscle (myo in Greek), the resulting impulses are translated into electrical signals. These EMG signals then drive prosthesis control using various decoding methods, including proportional control, on-off control, finite state machine-based control, and pattern recognition-based control, to name a few [6]. With each such method, EMG signals are detected using surface electrodes that are housed within the device socket, after which they are transmitted to the device's controller [6] (see Figure 1-1). For individuals with transradial amputations, electrodes can be placed over wrist flexor and extensor muscle sites of their residual limb, or an array of electrodes can be placed over all forearm muscles (as illustrated in Figure 1-1).

Controller software processes and interprets EMG data and transmits corresponding instructions to the device's battery-powered wrist or hand motors [7]. Rehabilitation clinicians, including prosthetists and occupational therapists, work with persons with amputation to select and assemble user-specific hand, wrist, controller, and EMG electrode components of a prosthesis, and to guide device fit and control learning for daily use [8].

Myoelectric device use can be challenging, even when advanced *pattern recognition-based control* is employed [7]. Such control is based on machine learning, which uses algorithms to establish relationships between input data and intended outputs (without explicitly programming these relationships for every input scenario). In myoelectric control, pattern recognition solutions learn and recognize patterns of EMG signals (those associated with predetermined muscle contractions) to predict and execute user-intended device movements [9]. Pattern recognition-based control has garnered much focus in myoelectric upper limb prosthesis control research [6]. Indeed, many experimental prosthesis controllers (along with an increasing number of commercial solutions) use this method.

To initialize a pattern recognition-based control algorithm (model), a device user must first perform a series of muscle contractions, known as a *training routine* [7]. For current pattern recognition-based control solutions, a training routine is usually performed in just one limb position; that is, with the user's arm at their side, elbow bent at 90° with hand in front [10]. Once EMG signal patterns are learned by the model through training, all subsequent signals captured during device use can be recognized for movement prediction purposes. Understandably, if a training routine is long, and/or if it must be re-executed to calibrate control throughout the day, users must expend a lot of time and focus. This problem is known as *training burden* [11], [12].

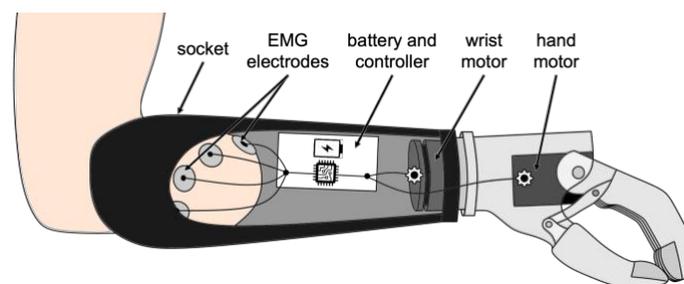


Figure 1-1. The inner workings of an example myoelectric prosthetic limb on an individual with transradial amputation. In this example, EMG signals are captured via an electrode array.

Although pattern recognition-based solutions offer users predictive device control after model training [13], unexpected changes in EMG signals can still cause control to degrade [14]. One major cause of these changes is known as the *limb position effect* problem [15], [16], [17], [18]. Here, EMG signals become unpredictable when a user's limb is used in a position different from that in which its control model was trained. Specifically, muscle activity changes when working against gravity, electrodes shift with respect to the muscles, and the topology of muscles vary [14]. In untrained positions, users often must struggle to maintain device control. An unintentional and potentially catastrophic drop of an object (a glass, for instance) might even result. Although this common problem is now well acknowledged, it remains largely unsolved [14].

To address the limb position effect problem, several studies have investigated augmenting EMG-based pattern recognition models with the addition of *contextual information*—specifically, information about a user's residual limb position during device use [14]. Some studies investigated capturing EMG data from multiple limb positions during model training routine execution, rather than from simply an elbow bent at 90° position. This approach yielded richer training datasets with improved device controllability outcomes [15], [16], [19], [20], [21]. Other studies found that information about a user's limb position could be captured using an inertial measurement unit (IMU) worn on their forearm. As an IMU provides linear acceleration signals on three axes, a device control model can use this data to learn about the user's limb in space. Indeed, outcomes from this experimental approach have shown improved model predictions [15], [16], [21], [22], [23]. Despite all such promising results, none of the controllers proposed by earlier studies have been translated from research to clinical applications, largely because prosthesis control research continues to focus on solutions that are simple to implement, require minimal data, and can coordinate sequential joint movements [24].

Capturing limb position data, in addition to EMG data, will create huge input datasets for a control model. This scenario poses a question: might some pattern recognition-based models be better suited to handle large amounts of data from different sources? This is not an easy question to answer, particularly as many machine learning models exist, including simpler classification- and regression-based solutions, along with more complex deep learning alternatives. For myoelectric prosthesis control, a simple classification algorithm, known as linear discriminant analysis, is traditionally employed [25], [26]. However, given the complexity of combining EMG and IMU

data, deep neural networks present another model approach. These networks can process large volumes of raw data and can form complex non-linear associations between input and output data. This capability enables such networks to tackle modelling challenges such as myoelectric prosthesis control. Neural networks, in fact, have been found to perform *better with more data* [27]. They are also capable of processing and learning from multimodal data [28]. As such, deep neural networks present a viable option towards the use of contextual information for *position-aware* myoelectric control solutions.

## 1.2. Research Objectives & Contributions

The objectives of the work undertaken in this thesis focus on exploring and developing myoelectric prosthesis control solutions for those with transradial amputation. The ***broad goal*** of the work is to offer users reliable, robust, and personalized device operation. Its ***specific objective*** is to develop a solution that improves overall prosthesis control while addressing the limb position effect problem. To address this unresolved problem, the work turns towards an exploration of deep neural networks, with an ***aim of contributing*** the following:

- ⇒ An algorithmic approach that leverages multimodal data and the power of deep learning to mitigate the limb position effect *and* allow simultaneous control of multiple prosthetic joints with varied velocities
- ⇒ Strategies for reducing training burden to improve controller usability
- ⇒ Appropriate metrics for comparative myoelectric prosthesis controller research aimed at solving the limb position effect problem

Ultimately, the work presented in this thesis bridges the gap between laboratory and real-use implementation of upper limb prosthesis control that uses pattern recognition. It presents an exciting new direction for myoelectric controllers of the future.

## 1.3. Thesis Outline

This thesis conforms to the University of Alberta's Faculty of Graduate & Postdoctoral Studies' paper-based format, with papers presented verbatim (that is, as they appeared in their published, accepted, or submitted format, but with some contextual changes introduced and grammatical

errors rectified). It is presented over a series of chapters, a summary of which is illustrated via a flowchart in Figure 1-2. The flowchart identifies the overarching objective of the work and includes specific research questions addressed, the publications resulting from that research, and subsequent research gap(s) that led to new questions. The thesis begins with a presentation of machine learning approaches for upper limb prosthesis control (Chapter 2). From there, more advanced methods of machine learning (recurrent convolutional neural networks, or RCNNs) are explored towards mitigating the limb position effect (Chapter 3). This exploration leads to a foray into an RCNN application with a modified model training paradigm (Chapter 4). As an alternative RCNN model training method, an RCNN solution with transfer learning (pre-training a model) is investigated and identifies a need for more comprehensive control assessment metrics (Chapter 5). This hurdle is overcome with the development of a novel set of metrics for comparative myoelectric prosthesis controller research (Chapter 6). With this toolset of metrics established, RCNNs are then more closely investigated to yield a feasible RCNN regression-based myoelectric control approach—a departure from commonplace classification-based approaches implemented to date (Chapter 7). The crowning accomplishment of this thesis is a case study that tested this RCNN regression-based solution using a person with transradial amputation (Chapter 8).

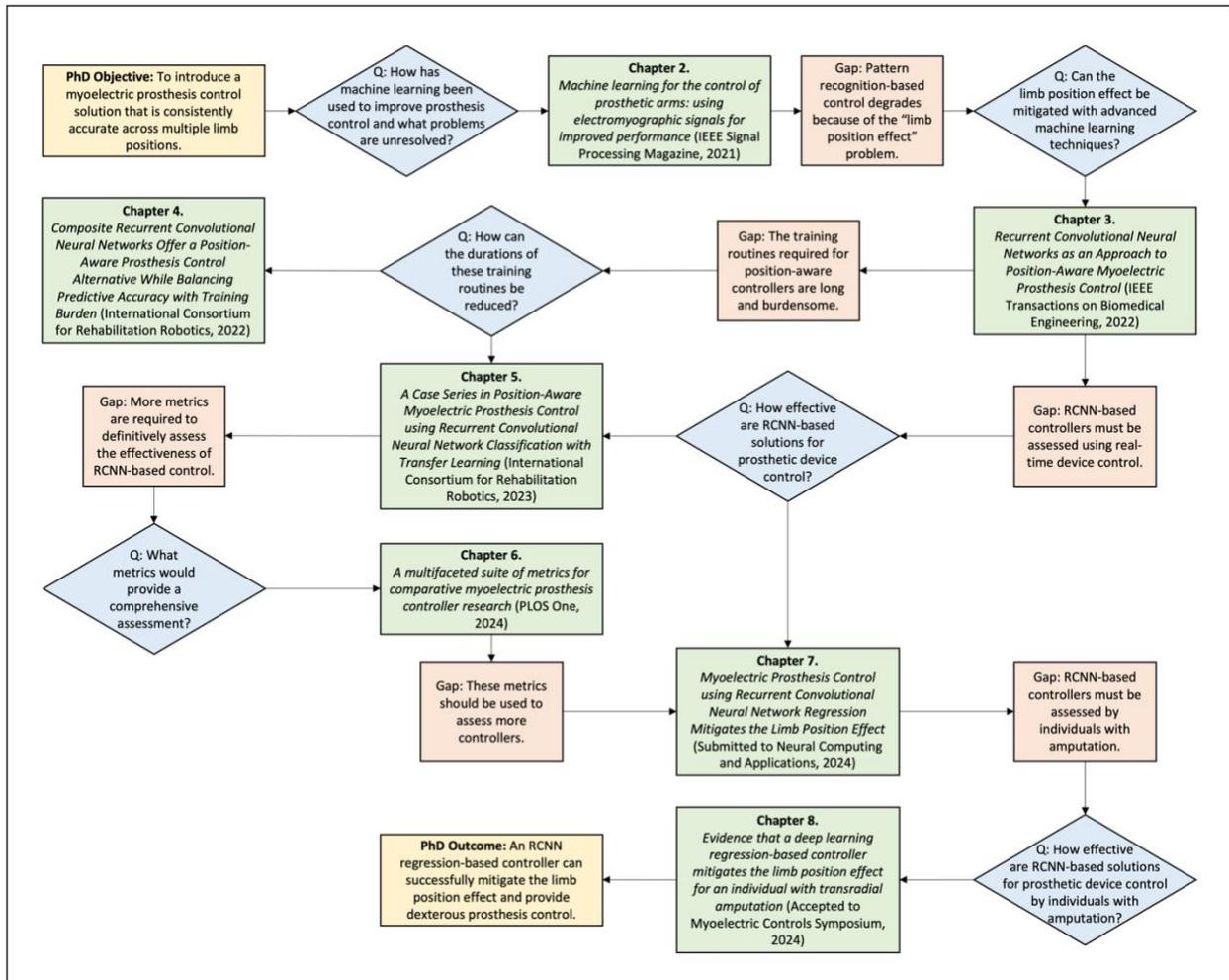


Figure 1-2. Summary of thesis chapters, with this work’s main objective and its outcome in yellow, research questions addressed in blue, chapter numbers with associated publications in green, and research gaps in orange.

# Chapter 2.

## Machine Learning for the Control of Myoelectric Upper Limb Prostheses

Chapter 2 presents “Machine learning for the control of prosthetic arms: using electromyographic signals for improved performance,” originally published in IEEE Signal Processing Magazine, in 2021 [13]. It begins by describing conventional myoelectric prosthesis control strategies and the use of electromyographic signals. Thereafter, it discusses a variety of control strategies that employ machine learning, each with their own benefits and limitations. Finally, it identifies the well-known but unresolved *limb position effect* problem, which should not be overlooked when investigating myoelectric control. An illustration of how the information presented in Chapter 2 contributes to the overall work in this thesis is presented in Figure 2-1.

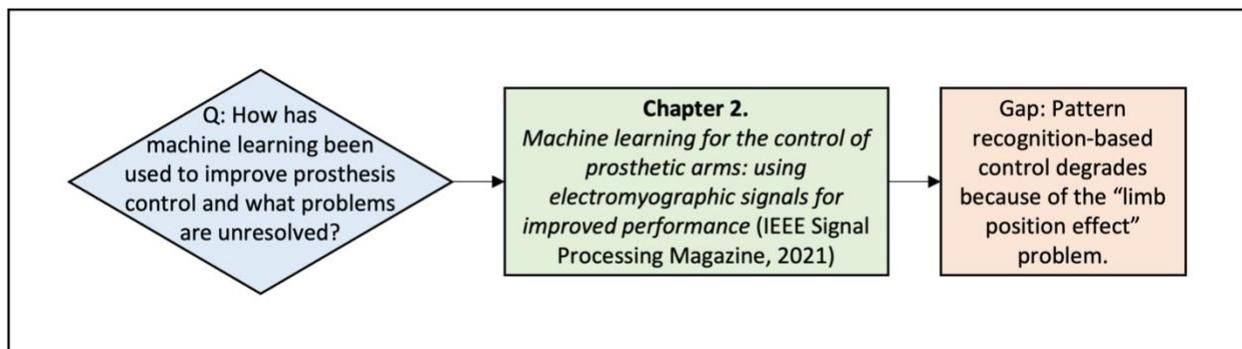


Figure 2-1. Research question addressed by Chapter 2 and remaining gap.

## Abstract

The human hand can perform many precise functions and is relied upon for countless aspects of daily life. When upper-limb amputation is necessitated, an affected individual's sense of independence is understandably impacted. Myoelectric prostheses are designed to restore some lost upper-limb motor function. Electromyographic (EMG) signals, generated by a user's muscle contractions in the residual limb and detected by electrodes in the device socket, activate prosthesis movement. A device controller processes these signals, decodes them, and then sends electrical instructions to device motors. Machine learning has been introduced to both decode and interpret useful features that exist in EMG signals so that improved motor control instructions can result. Although machine learning has been shown to yield more physiologically natural myoelectric prosthesis control for a user in research environments, it is not yet widely accepted by clinicians. In this work, we present an overview of how machine learning is used in EMG signal-driven upper-limb prosthesis control, along with a discussion about how it could be employed to improve the robustness and reliability of future devices.

### 2.1. Introduction

Upper limb prostheses are designed to restore lost hand and associated arm function for persons with amputation. Recreating dexterous hand function, however, poses a challenge as the upper limb is both complex in design and can perform coordinated movements across several degrees of freedom (DOFs). These DOFs include flexion/extension, forearm pronation/supination, fine finger joint movements, and more. Myoelectric prostheses are robotic devices (with motors, electronics, and control software) designed to mimic the movement and function of an anatomical arm and hand. A prosthetic device is connected to a user's residual limb via a custom-fabricated socket. Surface electrodes within the socket detect electromyographic (EMG) signals when the user intentionally contracts specific muscles in their residual limb. These muscle signals are then relayed to the prosthesis controller, which uses signal processing routines along with a control algorithm to filter this raw data, extract signal features from it, and determine the intended movement. Thereafter, the resulting control signals are mapped to electrical signals and the device motors carry out the instructions. This EMG signal-driven upper limb prosthesis control process is illustrated in Figure 2-2.

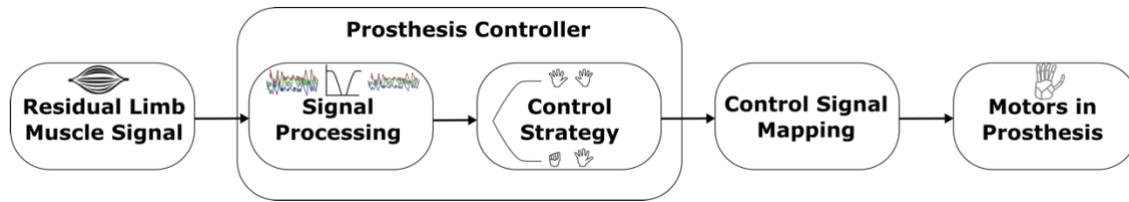


Figure 2-2. Overview of EMG signal-driven upper limb prosthesis control. EMG signals are detected by device socket electrodes, during specific muscle contractions in a user’s residual limb. The prosthesis controller processes and transforms the raw EMG signals using signal processing routines, and a selected control strategy algorithm determines the intended prosthesis movements. Further mapping is required to translate the control signals to electrical signals capable of driving motors in a prosthesis. Finally, motors are activated and deactivated by these electrical signals.

Decades of advancements in signal-driven prosthesis control have offered promising restorative movement options to persons with upper limb amputation [6]. Nevertheless, precise real-time decoding of movement intent from EMG signals has some notable limitations, and achieving accurate and intuitive prosthesis control remains a challenge [9].

### 2.1.1. Conventional Control Strategies

Conventional myoelectric prosthesis control strategies can be broadly categorized as either: (1) on/off, which enables binary opening and closing of a hand when EMG signals pass an activation threshold, or (2) proportional, which controls the speed of opening and closing of a hand, and thereby facilitates finer movement control. With either such control strategy, the actions of “hand open” and “hand close” are typically activated by contractions of two different residual muscles (with two different electrodes used to detect these opposing actions) [6]. Control of additional joint movements, then, requires the recruitment of other residual muscles. Although both on/off and proportional control strategies are considered robust [29], they are limited by the number of independent muscle signals in a user’s residual limb to control each DOF [7]. This limitation presents an operational challenge, given that advanced upper limb myoelectric prostheses typically offer more DOFs than the number of independent EMG signals that a device user can generate. In addition to this limiting factor, inherent variability in EMG signals can contribute to inconsistency in prosthesis control and lead to unintended prosthesis movements [14]. Variability in EMG signals can be introduced by changes in the user’s limb position, differences in the strength of their muscle contractions, fatigue, environmental factors, humidity, electrode shifting, and other within/between-day changes [14]. Despite offering functional dexterity to users, conventional

myoelectric prosthesis control strategies do not yet offer physiologically natural upper limb movements.

### 2.1.2. Machine Learning Control Strategies

To facilitate more dexterous and intuitive myoelectric device control, upper limb prosthesis researchers have been delving into the use of EMG signal-driven machine learning algorithms since the 1970s [30]. These algorithms have shown promise towards improving prosthesis control accuracy and ease of use for users, largely due to advances in signal processing techniques, computationally powerful processors, and enhanced battery technology. However, devices that employ machine learning algorithms are often not used in clinical settings because of their reported lack of robustness [29]. So, prosthesis researchers continue to investigate various EMG signal-driven device control strategies to overcome this limitation [29]. This work provides an overview of how EMG signal processing is undertaken in upper limb myoelectric prosthetic device control, introduces existing control strategies that use machine learning, and discusses the potential for machine learning to result in more physiologically natural myoelectric prosthesis control.

## 2.2. EMG Signal Processing Overview

Most commercially available upper limb myoelectric devices use open loop control strategies, wherein the device does not receive feedback from its environment [31]. Instead, a user is left to rely on visual feedback after deliberately contracting a muscle in their residual limb to initiate and sustain control of the device's movement. The electrical potentials resulting from this contraction generate raw EMG signals. These signals are typically sampled at a constant rate of between 200–1000 Hz (depending on the type of EMG electrodes embedded in the prosthesis socket). The raw signals are then processed, which includes filtering and the extraction of useful features.

### 2.2.1. Filtering

In general, raw EMG signals are low-pass filtered at 450 Hz to remove noise and high-pass filtered at 20 Hz to remove movement artifacts (originating at the electrode-skin interface) [32]. A comb filter is also often used to remove power line interference (at 60 Hz in North America, and at 50 Hz in Europe, Australia, most of Asia, and Africa), as well as to remove the harmonics of such interference [33].

### 2.2.2. Feature Extraction

Next, representative features are extracted from the filtered EMG signals, to ensure that the control strategy algorithm receives useful information. Representative features can generally be categorized as time-domain, frequency-domain, or time-frequency [34]. Time-domain features are most commonly used in both research and commercially available prosthesis controllers, and include Mean Absolute Value, Zero Crossing, Slope Sign Change, and Waveform Length [9]. Common frequency-domain features are Mean Frequency and Median Frequency [34]. The most common time-frequency representation is the Wavelet Transformation [14]. To accomplish representative feature extraction, filtered EMG signals are separated into blocks of data called windows and then for each window, features are calculated [35]. For example, windows may have a length of 160 milliseconds and may be overlapped with an offset of 40 milliseconds. After representative features are calculated for each window, they are ready for use as input to a myoelectric prosthetic device's control strategy algorithm.

## 2.3. Machine Learning in EMG Signal-Driven Prosthesis Control

Machine Learning is used in upper limb prosthesis control design to make devices more “intelligent”—that is, capable of predicting users’ intended movements. Myoelectric prosthesis researchers currently use various machine learning methods (based on elements of statistics and computing science), in the pursuit of more intuitive device control. Each method requires the development of a model that can be used to predict intended device movements, based on EMG signals captured during prosthesis use. A model, therefore, behaves as a mapping function that can take EMG signal input from socket electrodes (input channels) and map it to device motor instructions. To clarify machine learning terminology used herein, a “control strategy” refers to a type of model used in prosthetic device control, and an “algorithm” is a programmatic implementation of this strategy. Examples of control strategies presented in this section include Classification Control, Regression Control, Hybrid Control, Switching Control, and Movement-Based Control.

### 2.3.1. Classification Control Strategy

A classifier is a type of model that can be used in prosthetic device control. Given one or more input signals, a classifier will predict the value of one or more movement outcomes (via output

signals). The user must perform an extensive training routine to yield a training dataset that, in turn, is used to build the classifier. The classifier is then capable of predicting single or multiclass movement output signals. For example, a prosthetic hand that can open and close typically requires a classifier to be trained for 3 different movement classes: hand open, hand close, and no movement (rest). For each movement class, users are prompted to contract their residual muscles in a unique and consistent manner, mimicking the class of movement they are training. Once data for each class are recorded, a classifier can be built for use by the prosthetic device as its control strategy. Commonly used classification control strategy algorithms currently used in upper limb prostheses, are as follows:

- A) ***k*-Nearest-Neighbour** is a simple classification algorithm that forms a cluster from the training data to represent each class. Each cluster is identified as a *neighbour*. During classification, this algorithm uses a distance measure relative to the *k* (a positive integer) closest neighbors of a point to assign new data to a given class [36]. *k*-Nearest-Neighbour classifiers require only a short training time. The performance of this algorithm varies, depending on the number of chosen neighbours (forming coarse, medium, or fine connections), as well as the chosen feature set used to train the algorithm.
- B) **Linear Discriminant Analysis (LDA)** is an algorithm that reduces the dimensionality of features while maximizing class separability. For instance, a system that consists of 8 EMG channels and has 4 features extracted per channel, would yield (8x4) 32 dimensions (which may not be initially separable). This algorithm finds the least number of dimensions that could possibly be used to map input features to output classes, while maximizing the distance between each class cluster. LDA makes the restrictive statistical assumption that each trained class follows a Gaussian (normal) distribution [36]. This algorithm is most commonly used in research and commercial prosthesis control, as it requires minimal processing time and low training time while still providing high classification accuracy [9].
- C) **Support Vector Machine (SVM)** is an algorithm that attempts to maximize class separability by implicitly mapping input signal data into a high or even infinite dimensional feature space. The complexity of this algorithm varies depending on a chosen kernel [37], which is used to transform data into a desired feature space. Kernel functions that can be used by SVM [38] include:

- Linear kernel*, which assumes that all features are linearly related to the classes and finds linear coefficients that best fit the features with the trained classes' data. The shape of the decision boundaries developed by this kernel are straight lines or planes between the classes (Figure 2-3a and b).
- Polynomial kernel*, which models the relationship between the features and the trained movement classes as a nonlinear  $n^{\text{th}}$  degree polynomial. The shape of the decision boundaries developed by this kernel are curves or surfaces (Figure 2-3c).
- Radial Basis Function kernel*, which considers the Euclidean distance between each trained data point within a class and a computed centre between all the classes clusters. The Gaussian method is commonly used to compute this centre (Figure 2-3d).

Similar to the  $k$ -Nearest-Neighbour algorithm, SVM requires a greater training time than LDA, but may provide better classification accuracy than both algorithms.

D) **Artificial Neural Networks (ANNs)** have been used in myoelectric control for more than 30 years and are characterized by an increased capacity to form non-linear mappings between input channels and output classes [39]. In contrast to other algorithms discussed in this section, ANNs can learn complex relationships between input signals and desired outputs, all without the requirement for substantial inductive bias (such as assuming a

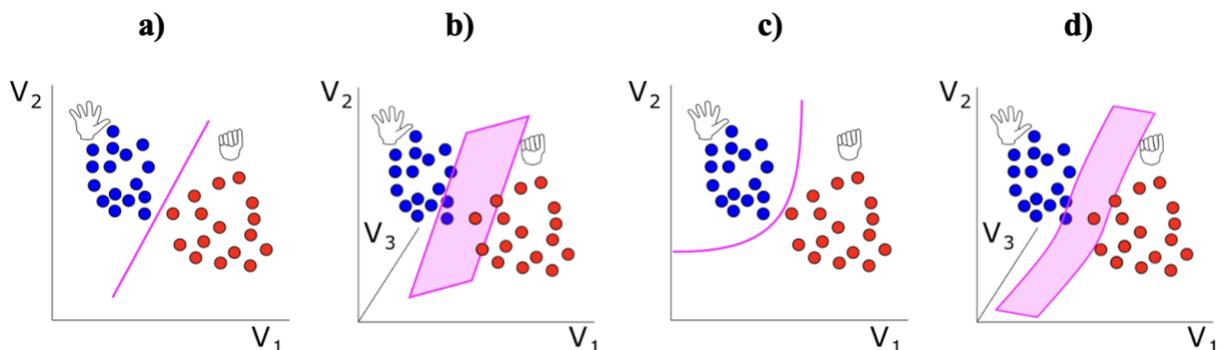


Figure 2-3. Visualization of class decision boundaries between hand open and hand close using LDA or SVM with a linear function kernel projected on a) a 2-dimensional vector space and b) 3-dimensional

vector space. Class decision boundaries using SVM with c) polynomial kernel and d) radial basis function kernel

mathematical relationship between inputs and outputs or using hand-selected extracted features). But, such generality and nonlinearity comes at the cost of the computational resources required for intensive training with large, balanced datasets (i.e. similar amounts of data for each class) necessary to prevent overfitting, forgetting, and other known issues [27]. Already a range of small consumer electronic devices, such as smartphones, now regularly deploy ANNs for their principal functions. Therefore, the use of ANNs is expectedly on the horizon for upper limb device control.

In summary, the number of EMG signals and the choice of the features set used to train an algorithm have a significant impact on the performance of a classifier [34]. One must consider the size of data used for training and the algorithm training time when selecting an algorithm. For instance, ANNs may provide higher performance for big data sets at the cost of higher training time. Conversely, LDA may provide lower performance but a significantly faster training time. The optimization of speed and performance may be achieved by using heuristics for selecting EMG signals, features, and feedback-based training protocols.

Classification-based myoelectric control strategies enable the prediction of single or multiclass movement output signals. Classifiers are only capable of classifying discrete movements sequentially and do not provide speed control. However, by implementing multiple classifiers that compare different combinations of movement classes, classification algorithms can offer simultaneous control at the expense of accuracy, but still without direct control over speed [40]. To overcome these drawbacks, regression control strategies are used in upper limb prosthesis research, as they enable both simultaneous and independent speed control.

### 2.3.2. Regression Control Strategy

A regressor is another type of model that can be used in prosthetic device control. Unlike classifiers that predict single or multiclass movement output signals, a regressor learns relationships between input signals and one or more continuous-valued movement output signals. Rather than training classes, regressors build models using data from DOF movements. Each DOF movement may include 2 or more classes: for example, hand open/no movement/hand close would be considered

1 DOF. In other words, a single DOF is mapped in a continuous space that traverses 2 or more discrete positions. For a prosthetic hand that can open, close, and rotate wrist, the number of training sets is reduced from 5 classes for a classifier (hand open, hand close, no movement, rotate wrist clockwise, and rotate wrist counter clockwise) to 2 DOFs for a regressor. In addition to this reduction, regressors trained for 2 DOFs enable the control of both DOFs simultaneously while not allowing the activation of multiple movements within a DOF (i.e., while an action of hand open and close may not occur simultaneously, hand open and rotate wrist clockwise may occur simultaneously).

Similar to classifiers, regressors require a training routine. This is known as dynamic DOF-based training and involves a series of special movements. For this training, users are prompted to slowly contract their muscles to move through the desired range of motion required by the regressor. Once data for each DOF are recorded and features are extracted, a regressor can be built, for use by the prosthetic device as its control strategy. Commonly used regression control strategy algorithms currently used in upper limb prostheses are as follows:

- A) **Multivariate Linear Regression** finds a relationship between several independent features and a target activation variable for a given DOF, while assuming that the relationship between these variables is linear [41]. Despite the simple implementation of this algorithm, the complexity of the dynamic input data poses a significant challenge to this linear relationship assumption. Figure 2-4a and b show a sample regressor output for 2 DOF trained data using a multivariate linear regressor.
- B) **Support Vector Regression (SVR)** has the same implementation as SVM but with a key difference — SVR uses dynamic DOF-based training data instead of class-based training data [42]. Figure 2-4c and d show a sample output for a 2 DOF trained data using SVR with a radial basis function kernel.
- C) **ANN Regression** has the same implementation as ANN classification, but uses dynamic DOF-based training data. As with ANNs for classification, regression training also often requires careful selection of large training datasets. ANNs have already been shown to be usable for regression control of myoelectric prostheses [43]. However, long training times may make current real-world prosthesis deployments of ANN-based EMG control

frustrating or intractable. However, as computation, actuation, and sensing component costs and performance improve through economies of scale, we expect ANN control strategies to show substantial gains in terms of clinically deployed performance in the next 2-5 years.

Although multivariate regression has a relatively simple implementation and is very fast to train, it is not recommended for control of more than 2 DOFs, given that its performance degrades significantly without implementing signal adaptation strategies and evoking activation thresholds. The procedure and prompts for recording training data from users affects the performance of these algorithms, since the dynamic change in signals and features drives the overall performance of the regressor.

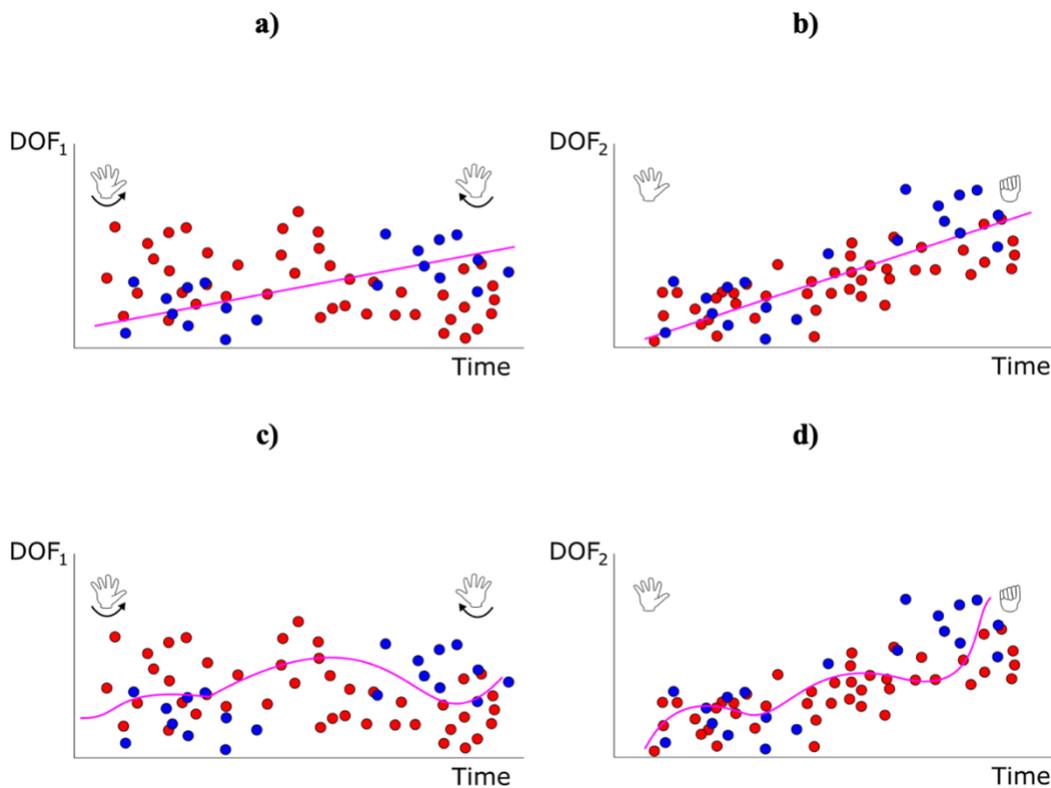


Figure 2-4. Visualization of regressor output over DOF 1 rotation and DOF 2 hand open/close using (a and b) multivariate linear regression where the algorithm finds 2 weights (one for blue feature and one for red feature) that may predict the transition from one hand position to the next within a DOF, and (c and d) SVR with a radial basis function kernel

In summary, regression-based myoelectric control strategies enable simultaneous and independent device speed control. However, this comes at the expense of the relative robustness of classifier-based control strategies that can manage unintentional muscle contraction variations introduced by users. Due to this lack of regressor robustness, classifiers are more commonly implemented in upper limb prosthetic devices [6].

### 2.3.3. Hybrid Control Strategy

To mitigate the trade-off between increased control and decreased robustness when regression control strategies are solely used in upper limb device control, researchers have opted to combine classification and regression control strategy approaches. That is, SVM algorithms have been used for classifying DOF training data, with SVR algorithms used for DOF speed regulation [44]. This approach has reportedly resulted in improved intended device performance, yet at the expense of a long training time durations for all limb positions controlled (due to a large amount of training data) [44].

An alternative hybrid control strategy that is anticipated to yield better control performance across various limb positions involves integrating inertial measurement unit (IMU) data in 3-dimensional space. This approach has been implemented using two different methods. One method is to provide a single classifier with both IMU data and EMG features [15], as shown in Figure 2-5a. The other is through the use of a classifier-classifier cascade, in which IMU data are used to classify the limb position, and then the limb position classification is used to select the appropriate EMG classifier [15]. Figure 2-5b shows this cascade method. These methods have already been shown to improve EMG classification accuracy across multiple limb positions [14]. We predict that deep learning may further improve this accuracy, given that it has been shown to handle multi-modal data [27]. Although both methods have been shown to improve the accuracy of EMG classification when various arm positions are introduced, we additionally propose that a “Position Regression – Movement Classifier” cascade would likely further improve this accuracy, given that arm positions are not discrete. Although such a regression-classifier cascade will demand higher processing power than a classifier-classifier cascade, it may require a shorter training time as compared to a regression-regression cascade.

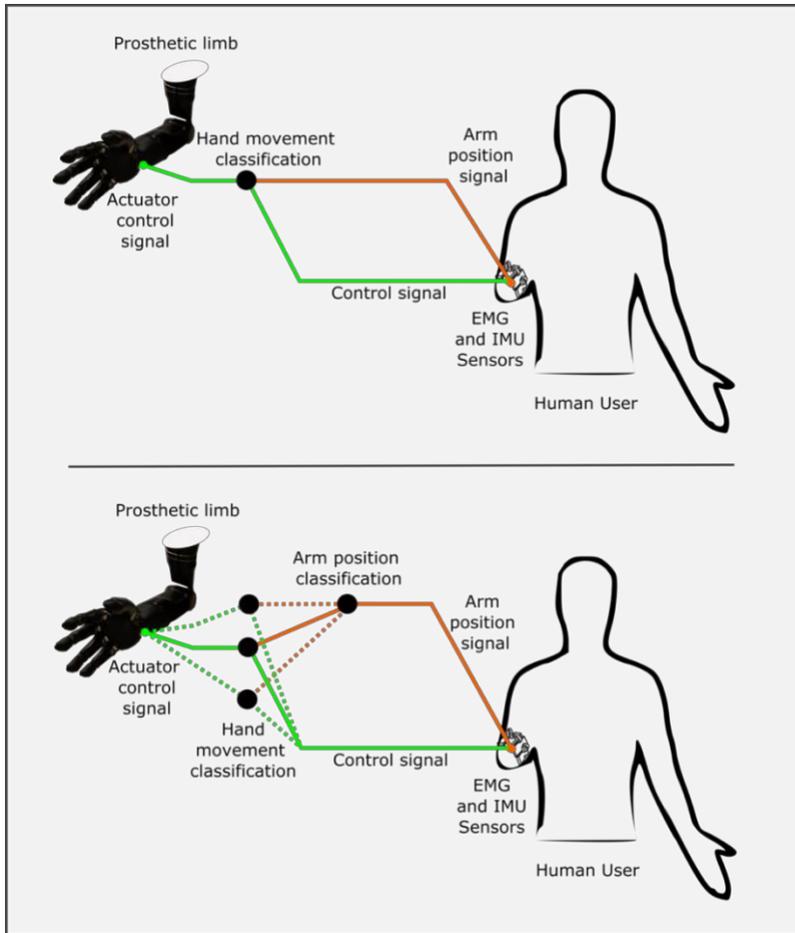


Figure 2-5. Hybrid control requiring 2 input signals to the prosthetic limb—an arm position signal and a control signal, with either a single classifier a) or a classifier-classifier cascade b).

Overall, as additional data are required to implement hybrid control strategies, any resulting improvement in device performance tends to be coupled with increases in algorithmic complexity, computational time, and hardware costs. Therefore, the choice of a control strategy for an upper limb prosthesis should not be entirely assessed based on performance but rather on a trade-off between computational burden and performance (especially in embedded systems operating with limited processing power and memory).

#### 2.3.4. Switching Control Strategy

Switching (or gated) control is another strategy that has been integrated with conventional myoelectric prosthesis control methods. This control strategy closes the gap between the number of DOFs offered by an advanced myoelectric device and the limited number of EMG signals that

a device user can physically generate using their residual limb muscles. In switching control, the device user can switch between activating multiple device DOFs (specifically, joint movements or grasp patterns) using muscle co-contraction or an external mechanical toggle (Figure 2-6). Although this method offers sequential control of multiple DOFs, it is reportedly slow and unintuitive for a user [7]. A subset of machine learning, known as reinforcement learning, has been investigated as an improvement to adaptive switching [45]. In this control strategy, the list of possible DOFs is continually reordered while a prosthesis user performs tasks, based on the predicted likelihood of each DOF being used next [45]. This strategy has been shown to decrease the number of required user switches, and the time necessary to complete tasks [45]. Autonomous switching extends the methods developed by adaptive switching. It further decreases the number of switches that the device user must trigger, as it uses predictions to switch between DOFs automatically [46]. Despite such advances towards control of multiple DOFs, switching methods of all forms remain somewhat unintuitive to users, and as such, have not frequently been explored in upper limb prosthesis research.

### 2.3.5. Movement-Based Control Strategy

Movement-based control, also known as synergistic or trajectory control, is another strategy that addresses the challenge of limited residual muscle DOF control for device users. The movement-based control strategy requires that a user's intact joint movement be measured using IMUs (such as the shoulder position). These measurements are then used to drive the movement of a prosthetic

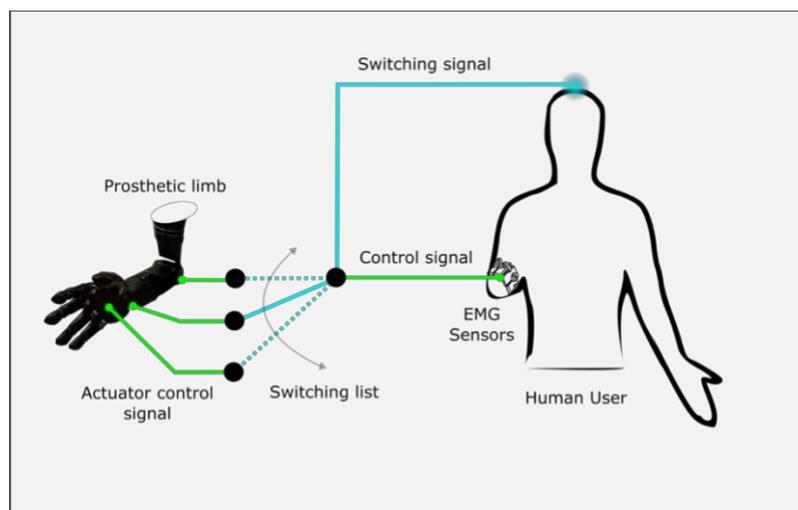


Figure 2-6. Switching control requiring 2 input signals to the prosthetic limb—a switching signal and a control signal.

joint (such as the elbow) [47]. This strategy is advantageous, as it reduces the number of device DOFs that require myoelectric control and instead allows control to be isolated to more distal DOFs, such as hand open/close. Movement-based control is based on known physiological synergies between joints (established using artificial neural networks [47], or simply mathematical relationships between measured and controlled joint angles [48]). For instance, shoulder movements have been shown to synergistically drive elbow flexion/extension [47], as well as wrist rotation [48]. Movement-based control has been shown to yield prosthetic arm movements that perform closer to unimpaired ones, reduce compensatory movements that are common to prosthesis users [47], and reduce task performance time (as compared to conventional sequential control) [48]. Furthermore, movement-based control accuracy is reportedly improved when paired with contextual information [49] or upper arm EMG signals [50].

## 2.4. Conclusion and Future Direction

This work has presented several EMG signal-driven upper limb prosthesis control strategies that use machine learning. These control strategies collectively demonstrate that machine learning approaches can yield more accurate prosthesis control over conventional (on/off or proportional) alternatives. However, it has not yet been determined which of these approaches is most favorable in the pursuit of a truly “intelligent” robotic replacement limb design—one that adapts to its environment to achieve the user’s desired movement in real-time.

Ideally, an intelligent prosthetic device should be able to integrate EMG signals and close the control loop within the prosthesis controller using captured sensory data and within the human-prosthesis system by providing the user with sensory feedback. Although closing this control loop remains a challenge, prosthesis researchers have explored addressing this challenge by restoring the tactile feedback of human touch via tactors [51] or implantable electrodes [31], and augmenting it with biofeedback [52]. As a consequence, users reportedly feel more connected to their devices [31]. Although myoelectric prostheses represent the clinical state of the art, current commercial devices do not intentionally provide sensory feedback [53]. Therefore, closing the control loop within the controller becomes more appealing.

Given that the fingers of some prosthetic hands have the capability of detecting pressure [54], the corresponding pressure data could be used by a device controller during a grasp action to halt hand

closure. This automatic action may contribute to reducing cognitive load for operating the device. Furthermore, closing the control loop through the provision of both user environment and task context will ultimately contribute to device performance improvements. For example, if an upper limb prosthesis user could train different device control algorithms for specific environments or daily routines (such as preparing breakfast in a kitchen or checking emails on a computer), a smarter controller would undoubtedly result. We predict that in the near future, novel device control algorithms that learn myoelectric prosthesis control in real-time using deep neural networks can be anticipated [55]. These methods are expected to offer transformative capabilities in delivering contextual device control with multiple sensory inputs (EMG, IMU, and others), and therefore greater clinical and user acceptance. Ultimately, it is our hope that soon, fully regained dextrous upper limb motor function can be offered to persons with amputations through the use of truly “intelligent” prosthetic devices.

# Chapter 3.

## Recurrent Convolutional Neural Networks (RCNNs) for Position-Aware Control of Myoelectric Upper Limb Prostheses

Chapter 3 presents “Recurrent Convolutional Neural Networks as an Approach to Position-Aware Myoelectric Prosthesis Control,” originally published in IEEE Transactions on Biomedical Engineering, in 2022 [10]. It considers how deep neural networks can be used for prosthesis control, in response to the limb position effect problem identified in Chapter 2. In this current chapter, neural network-based control models (both classifiers and regressors) are compared to commonly used control models (that do not use deep learning techniques). Two important lessons stemming from this work include the understanding that: (1) model training routines can result in user training burden, and (2) off-line control assessments might not represent real-use control experiences. An illustration of how the information presented in Chapter 3 contributes to the overall work in this thesis is presented in Figure 3-1.

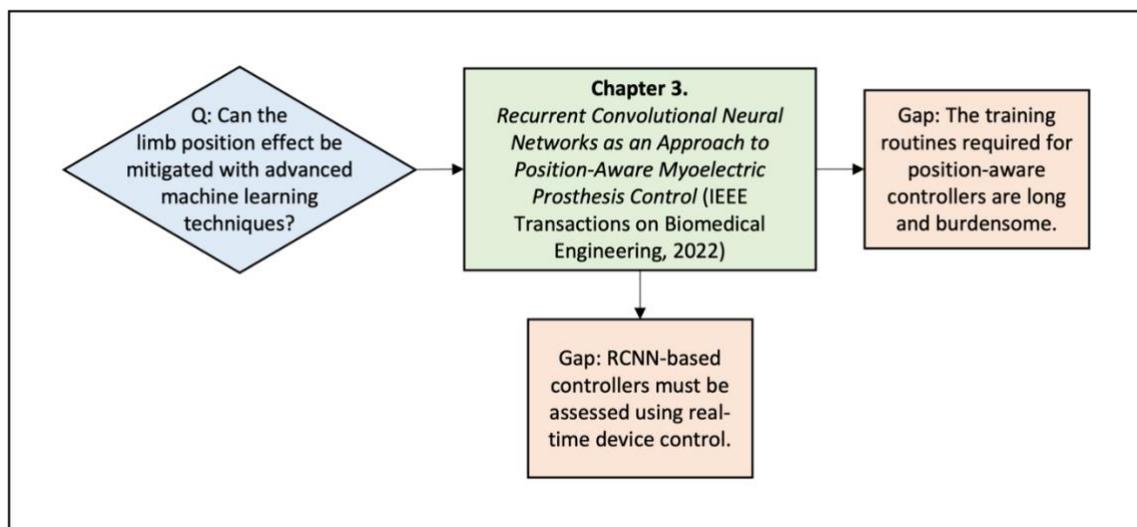


Figure 3-1. Research question addressed by Chapter 3 and remaining gaps.

## Abstract

*Objective:* Persons with normal arm function can perform complex wrist and hand movements over a wide range of limb positions. However, for those with transradial amputation who use myoelectric prostheses, control across multiple limb positions can be challenging, frustrating, and can increase the likelihood of device abandonment. In response, the goal of this research was to investigate convolutional neural network (RCNN)-based position-aware myoelectric prosthesis control strategies. *Methods:* Surface electromyographic (EMG) and inertial measurement unit (IMU) signals, obtained from 16 non-disabled participants wearing two Myo armbands, served as inputs to RCNN classification and regression models. Such models predicted movements (wrist flexion/extension and forearm pronation/supination), based on a multi-limb-position training routine. RCNN classifiers and RCNN regressors were compared to linear discriminant analysis (LDA) classifiers and support vector regression (SVR) regressors, respectively. Outcomes were examined to determine whether RCNN-based control strategies could yield accurate movement predictions, while using the fewest number of available Myo armband data streams. *Results:* An RCNN classifier (trained with forearm EMG data, and forearm and upper arm IMU data) predicted movements with 99.00% accuracy (versus the LDA's 97.67%). An RCNN regressor (trained with forearm EMG and IMU data) predicted movements with  $R^2$  values of 84.93% for wrist flexion/extension and 84.97% for forearm pronation/supination (versus the SVR's 77.26% and 60.73%, respectively). The control strategies that employed these models required fewer than all available data streams. *Conclusion:* RCNN-based control strategies offer novel means of mitigating limb position challenges. *Significance:* This research furthers the development of improved position-aware myoelectric prosthesis control.

### 3.1. Introduction

Myoelectric prostheses are designed to restore lost upper limb motor function for individuals with amputation. Recreating the coordinated movements of a natural human wrist and hand, however, remains a challenge for those with transradial amputations. In response, researchers have developed control strategies that use pattern recognition models to predict and execute a user's movement intent [7]. Electromyography (EMG) is currently the most commonly used input source for prosthesis control [6], whereby EMG signals generated by muscle contractions in a user's

residual limb are captured by electrodes embedded in a device socket. Despite yielding reliable device movements in research environments, precise decoding of movement intent from EMG signals can be unreliable when a wide range of limb positions are introduced by users during daily activities [14].

This significant challenge to myoelectric prosthesis control is known as the *limb position effect* [18]. Often, detected surface EMG control signals are altered when a user's limb is in a position different from that in which the prosthesis controller was trained (usually a comfortable, low position) [18]. Resulting EMG signal variations can cause prosthesis control to degrade and unexpected prosthetic wrist and hand movements to occur. Researchers have investigated various methods of mitigating this problem, including the use of intramuscular electrodes [56], [57], high-density surface electrode arrays [58], [59], and wearable limb position sensors [15], [16], [17], [21], [22], [23]. However, a solution that provides reliable and practical position-aware control has yet to be found. As such, continued research is required.

Various pattern recognition approaches have been explored to address the limb position effect on end effector control [15], [16], [19], [20], [21], [22], [23], [60], [61], [62], [63], [64], [65], [66]. Broadly, pattern recognition approaches have included Statistical Models and Neural Networks (including deep learning), each of which can use either classification or regression techniques [14], [61]. Typically, classification models (classifiers) and regression models (regressors) map EMG signal features, which are extracted from raw EMG data, to predict intended end effector movements [34]. Classifiers map signal features to one of a *discrete* set of known classes (categories) of degrees of freedom (DOFs), offer control over multiple DOFs, but do not provide proportional control over device movement velocity or simultaneous control over multiple DOFs [6]. Conversely, regressors can map signal features to *continuous* velocity values for each DOF (proportional to input signal strength), offer simultaneous control of separate DOFs [6], but tend to be less robust than classifiers due to the increased complexity of their predictions [30]. Whether classification or regression is used for control, all models require a device training routine to be undertaken by the user, to inform pattern learning [67]. Although more training data generally yields stronger models, long training routines are cumbersome for the user [6], [12]. Overall, not only does the chosen pattern recognition model influence the resulting device control, the duration

of its required training routine, the time needed to train the model and make predictions, and the complexity of the model algorithm are also considerations.

Statistical models apply probability theory to learn patterns in data and are currently more often employed in position-aware prosthesis control research than deep neural network alternatives [3]. Some researchers have collected EMG data across multiple limb positions to inform statistical classifiers [15], [16], [19], [20], [21], while others have added positional information (quaternions or accelerometer data) to take limb orientation into account [15], [16], [21], [22], [23]. Statistical regressors have not been as extensively explored as classifiers in device control literature [30]. Nevertheless, both statistical classifiers and regressors offer the benefits of being straightforward to implement and having low computational costs [34]. Each such model requires representative feature extraction from EMG signals. This means that assumptions must be made regarding which features best inform movement prediction [34], [68], [69]. To avoid possibly making ill-informed or erroneous assumptions, researchers have also begun to explore the benefits of neural network methods for EMG-based control [27], [65].

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) may yield improved prosthetic movement prediction accuracies over statistical approaches, given the advantages that they offer. The first such advantage is that CNNs can predict intended movements from raw EMG signals (rather than from extracted features) [43], [70], [71]. This means that, given sufficient data, new features can be automatically learned, thereby avoiding the need for feature engineering. Another advantage is that CNNs offer the ability to combine a high volume of data from multiple sensors [27], [28]. This suggests that CNNs may prove to be effective towards learning the complex features of combined EMG and inertial measurement unit (IMU) signals across multiple limb positions. Furthermore, as time-domain features are commonly used for prosthesis control [34], recurrent neural networks (RNNs), which leverage the temporal behaviour of signals [72], might also be beneficial towards solving the limb position effect. Given that recurrent convolutional neural networks (RCNNs) can harness the collective advantages presented by CNNs and RNNs, they too offer a promising research direction for improving device control.

Compared to statistical approaches, few studies have explored using RCNN or CNN-based models for prosthesis control. Xia et al. examined the use of RCNNs, with raw EMG data, for the prediction of shoulder position (irrespective of end effector function) [72]. Their proposed model

yielded higher predictive accuracy than an alternative statistical regressor (support vector regression, SVR). Amongst other things, this research demonstrated that an RCNN can indeed learn features from raw EMG data to inform limb position. Ameri et al. confirmed that a CNN can be used with raw EMG data to predict wrist movement, and yielded offline and real-time performances better than those of an SVR [43]. More recently, Bao et al. used an RCNN to extract EMG features for the prediction of wrist motion [73]. This approach outperformed CNN-only approaches during complex wrist movements, and further supports the predictive potential of RCNN models. In 2018, Phinyomark and Scheme reviewed the potential for developing more advanced applications of EMG pattern recognition using deep learning approaches [27]. Collectively, the abovementioned studies recommended the continued pursuit of deep learning, including combining CNNs with RNNs, further optimizing model architectures, conducting more online testing of such models, or testing with larger datasets [43], [72], [73].

The goal of this study was to investigate the novel use of position-aware RCNN-based myoelectric prosthesis control strategies, towards solving the *limb position effect* problem. To this end, this study examined device control strategies that: (1) combined EMG and IMU input data streams to inform prosthesis movements and limb positions, respectively; and (2) used RCNN models to make movement predictions from these data. For each RCNN model under investigation, resulting movement predictions were compared to those of commonly used statistical models, so that potential improvements could be ascertained. The criteria by which position-aware control strategies were evaluated included their movement prediction accuracy, along with the number of EMG and/or IMU data streams that they required. Based on these criteria, this study identified two promising RCNN-based myoelectric prosthesis control strategies that were found to be consistently accurate across multiple limb positions.

## 3.2. Methods

### 3.2.1. Participants

A total of 19 participants with no upper-body pathology or recent neurological or musculoskeletal injuries were recruited. The data from 3 participants were incomplete and as such not used for this study. Of the remaining 16 participants, 3 had previous experience with EMG control, all had normal or corrected to normal vision, 15 were right-handed, 8 were male, and the mean age was

26.4 ± 8.7 years (± 1 standard deviation). Each participant provided written informed consent, as approved by the University of Alberta Health Research Ethics Board (Pro00086557).

### 3.2.2. Experimental Setup

Two Myo gesture control armbands (Thalmic Labs, Kitchener, Canada) were used to collect EMG and IMU data. Each armband contained 8 surface electrodes and an IMU. Each surface electrode collected 1 EMG data stream (sampled at 200 Hz). Each IMU collected 10 limb position data streams (3 accelerometer, 3 gyroscope, and 4 quaternion, all sampled at 50 Hz). Using Myo Connect software, the EMG and IMU data were streamed into Matlab. Hardware and software limitations required that each Myo armband be connected to a separate laptop, so two Lenovo ThinkPad laptops were employed. A custom Matlab script, running on one laptop, captured streamed data from one Myo armband and simultaneously displayed onscreen instructions for a participant to follow. At the same time, another custom script ran on the second laptop to record data from the second Myo armband.

### 3.2.3. Data Collection

Each participant donned two Myo armbands on their self-identified dominant arm, as shown in Figure 3-2A. One was worn on their forearm, with a mean distance of 6.0 ± 1.9 cm distal to the olecranon, and *electrode 1* on the lateral side of their forearm. The second armband was worn on

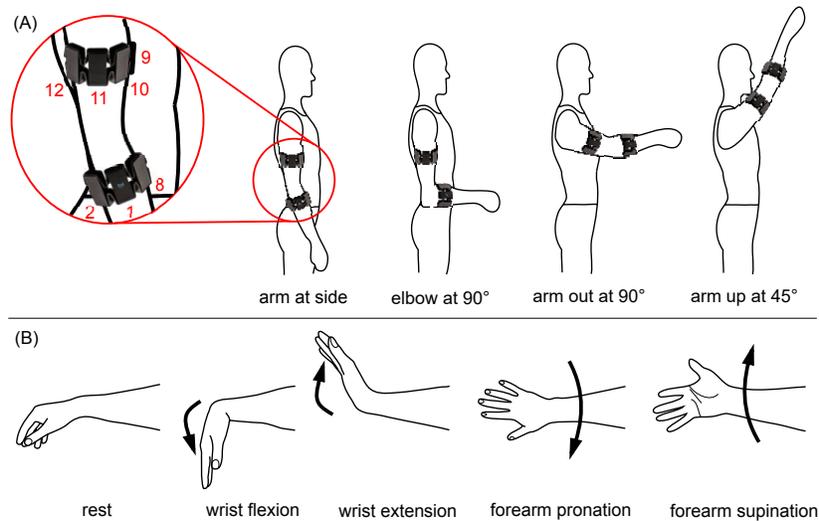


Figure 3-2. (A) The placement of each Myo armband, along with limb positions used in data collection, and (B) movements used in data collection.

their upper arm, with a mean distance of  $12.0 \pm 2.4$  cm proximal to the olecranon, and *electrode 9* on the anterior side of their upper arm, over the biceps muscle.

Participants followed onscreen instructions, performing various movements in 4 limb positions, as described below.

- **Movements** included: rest (relaxed), wrist flexion, wrist extension, forearm pronation, and forearm supination, as shown in Figure 3-2B. These movements are functionally important for individuals with transradial amputation [74]. Notably, the hand open and close movements were not included in this study, given that wrist flexion and extension can instead be used to control the opening and closing of a *prosthetic* hand [43], [53], [75], [76], [77]. Similarly, forearm pronation and supination can be used to control *prosthetic* wrist rotation.
- **Limb positions** included: arm at side, elbow bent at  $90^\circ$ , arm out in front at  $90^\circ$ , and arm up at  $45^\circ$  from vertical, as shown in Figure 3-2A.

Data collection consisted of 6 trials: 3 static trials and 3 dynamic trials. Rest time was provided between each trial.

- Trials 1–3 (**static**) required participants to perform various movements (shown in Figure 3-2B) using sustained isotonic muscle contractions. All movements were held for 5 seconds, separated by 5 seconds of rest. The movements were repeated in each of the 4 limb positions (shown in Figure 3-2A). Participants were instructed to perform each muscle contraction at a moderate effort that could be sustained for 5 seconds.
- Trials 4–6 (**dynamic**) required participants to perform movements that oscillated either between wrist flexion and extension or forearm pronation and supination. The timing of these oscillations was demonstrated onscreen (5 cycles with a period of 4 seconds). These oscillations were repeated in each of the 4 limb positions (shown in Figure 3-2A).

#### 3.2.4. Data Pre-Processing

The EMG data from each Myo armband were filtered using a high pass filter at 20 Hz (to remove movement artifacts), as well as a notch filter at 60 Hz (to remove electrical noise). Then, the IMU

data streams were resampled to 200 Hz using linear interpolation to align them with the corresponding EMG data. The resulting data from the two Myo armbands were synchronized.

The static trials were segmented into movements (rest, wrist flexion, wrist extension, forearm pronation, and forearm supination). For the dynamic trials, target sinusoids were generated to represent movement oscillations. Given that an offset was evident between participants' movements and onscreen oscillations, their sinusoids were corrected as follows: forearm EMG signal peaks were identified and used to fit a sine wave to represent wrist flexion/extension oscillations, whereas forearm IMU signal peaks and valleys (specifically from the accelerometer) were used to fit a sine wave to represent forearm pronation/supination oscillations. The resulting target sinusoids were then used to segment the dynamic trials into movements in each DOF (wrist flexion/extension, forearm pronation/supination).

Next, for the purposes of the RCNN models under investigation and their comparative statistical models, data were segmented further into windows (160-millisecond with a 40-millisecond offset). For the statistical models, time-domain features were then calculated for each EMG or IMU channel, in each window. These included 4 commonly-used EMG features (mean absolute value, waveform length, Willison amplitude, and zero crossings [9]) and 1 IMU feature (mean value). For the RCNN models, time-domain features were not calculated and instead, filtered signal data remained in each window.

### 3.2.5. RCNN and Comparative Statistical Models

#### 3.2.5.1. RCNN Models – Architecture

Matlab software was used to implement the RCNN models. Bayesian optimization automatically determined the number of convolution layers, number of filters, filter size, pooling size, and patience required in this study. Optimization was performed in two steps: first, the number of layers along with each hyperparameter being optimized were determined using a broad range of values; thereafter, values were refined using a narrower range (centered at earlier optimized values).

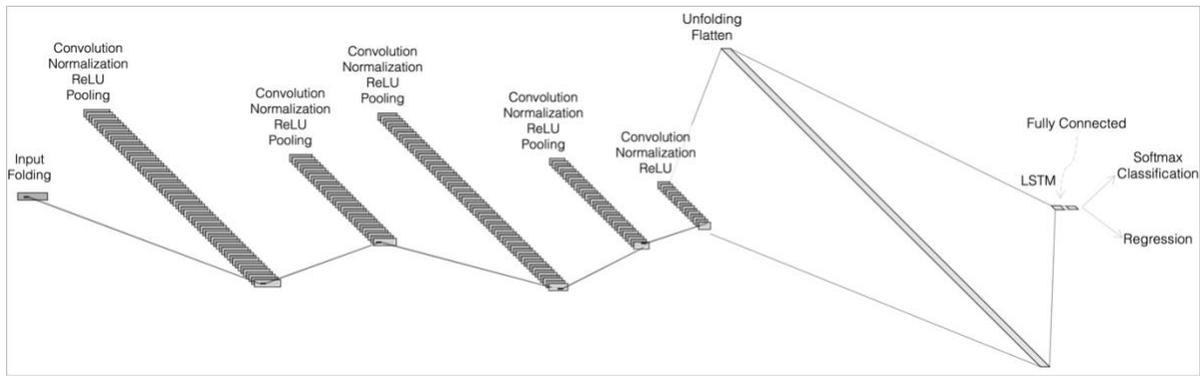


Figure 3-3. RCNN architecture: sequence input layer (Input); sequence folding layer (Folding); 4 blocks of convolution, batch normalization (Normalization), rectified linear unit (ReLU), and maximum pooling (Pooling); 1 block of convolution, batch normalization, and ReLU; sequence unfolding layer (Unfolding); flatten layer; long short-term memory (LSTM) layer; fully connected layer; and finally either (1) softmax and classification layers, or (2) a regression layer. Figure made using NN-SVG [78].

Our resulting RCNN models had architectures that consisted of 27 (classification) or 26 (regression) layers, as shown in Figure 3-3 [78]. In these models, a sequence input layer first received and normalized the training data. Then, a sequence folding layer was used, allowing convolution operations to be performed independently on each window. This was followed by a block of 4 layers: a convolution, a batch normalization, a rectified linear unit (ReLU), and a maximum pooling layer. This block was repeated 3 more times. Each of the 4 maximum pooling layers had a pooling size of 1x2. A block of 3 layers followed: a convolution, a batch normalization, and a ReLU layer.

- For **limb position classification**, the optimal number of filters in the convolution layers were determined to be 32, 32, 32, 64, and 64, respectively, and each had a filter windows size of 1x3.
- For **movement classification and regression**, the optimal number of filters in the convolution layers were determined to be 64, 32, 64, 32, and 16, respectively, and each had a filter window size of 1x5.

Subsequent layers included a sequence unfolding layer (to restore the sequence structure), a flatten layer, a long short-term memory (LSTM) layer, and a fully connected layer. Finally, either (1) a

softmax layer and classification layer were used, or (2) simply a regression layer was used. To prevent overfitting, a patience parameter was set that triggered early stopping.

### 3.2.5.2. Comparative Statistical Model

Given that linear discriminant analysis (LDA) is commonly used in prosthesis control research [15], [22], [23], this study opted to use LDA classifiers for comparisons to both RCNN limb position classifiers and RCNN movement classifiers. The chosen LDA discriminant type was pseudo-linear, since columns of zeros were occasionally present in rest classes for some features (including Willison amplitude and zero crossings).

SVR regressors were used for comparisons to RCNN movement regressors, as per earlier research [43], [72]. The SVR regressors used a linear kernel for input data mapping, given that it yielded the most accurate movement predictions in earlier pilot work (compared to radial basis function and polynomial kernel alternatives). This pilot work was based on EMG and IMU data from multiple limb positions. The kernel scale parameter was automatically optimized by Matlab software and no kernel offset was used.

### 3.2.6. Classification and Regression

This study explored models that predicted limb positions and movements. Three model specifications (S1–S3) were investigated, in addition to a comparative baseline model. All model specifications were substantiated by earlier research:

S1 – Model trained with EMG data from all limb positions [15], [16], [19], [20], [21]

S2 – Model trained with EMG and IMU data from all limb positions [15], [21], [22], [23]

S3 – Models trained with EMG data at each limb position, with subsequent predictions occurring in a **2-staged sequence**: 1<sup>st</sup>, a limb position was classified using IMU data; 2<sup>nd</sup>, a corresponding model (trained at that specified limb position) predicted a wrist movement using EMG data [15], [16], [22], [23]

Baseline – Model trained with EMG data from arm at side

Note that S1 and the Baseline require only EMG data, whereas S2 and S3 require both EMG and IMU data. Specifications S1–S3 and the Baseline were each implemented using an RCNN

classifier, an RCNN regressor, an LDA classifier, and an SVR regressor (16 models total). The training and testing of each model were performed in Matlab using an Intel® Core™ i9-9900K CPU (3.60 GHz).

### 3.2.6.1. Limb Position Classification

RCNN limb position classifiers were compared to LDA limb position classifiers in this study. The RCNN classifier inputs were signals from each window and the LDA classifier inputs were time-domain features from each window. Both classifiers outputted a predicted limb position class (shown in Figure 3-2A) for each window. Limb position classifiers were trained with Trials 1–2 (static) data from a participant and subsequently tested using Trial 3 (static) data from that same participant. This approach was motivated by current myoelectric prosthesis use, wherein the user must train their device controller before it can predict movement intent.

Prosthesis control research has shown that the use of numerous data streams (EMG and/or IMU) can result in longer machine learning processing times and/or increased hardware costs [22]. Taking these drawbacks into consideration, the specific data stream *types* that would most accurately inform limb position were initially investigated. Data from both Myo armbands were used in this investigation, with the RCNN and LDA limb position classifiers trained and tested using the following data stream combinations:

- All EMG and IMU data streams from both Myo armbands
- All EMG data streams from both Myo armbands
- All IMU data streams (quaternions, gyroscope, and accelerometer) from both Myo armbands
- Only accelerometer data streams [15], [16], [62] from both Myo armbands

Note that gyroscope and quaternion data streams were not investigated independently. Earlier pilot work revealed that accelerometer data better informed limb position in comparison to gyroscope and/or quaternion data.

### 3.2.6.2. Movement Classification

RCNN movement classifiers were compared to LDA movement classifiers. As with limb position classification, the RCNN movement classifier inputs were signals from each window and the LDA classifier inputs were time-domain features from each window. Both the RCNN and LDA movement classifiers outputted a predicted movement class (shown in Figure 3-2B) for each window. Movement classifiers were trained with Trials 1–2 data from a participant, and subsequently tested using Trial 3 data from that same participant.

Movement classifiers were trained, tested, and compared under model specifications S1–S3. The predictive accuracies of these classifiers were compared to those of a baseline classifier (BC), trained with only EMG data collected with each participant’s arm at their side (as per standard prosthesis training). Additionally, to minimize the number of data streams necessary for movement classification, each classifier was trained with the following combinations:

- Data (EMG and, when applicable, IMU) from only the forearm Myo armband
- Data from both Myo armbands
- EMG data from the forearm and IMU data from both Myo armbands (when applicable)

### 3.2.6.3. Movement Regression

RCNN movement regressors were compared to SVR movement regressors. The RCNN and SVR movement regressors used the same inputs as did the RCNN and LDA movement classifiers, respectively. However, the RCNN and SVR regressors outputted *continuous* movement predictions, denoting muscle activation intensity for each DOF (flexion/extension and pronation/supination) in each window. DOF range *endpoints* included:

- full flexion = -1, full extension = 1
- full pronation = -1, full supination = 1

Within each DOF range, 0 indicated rest. Notably, a single RCNN regressor was capable of yielding movement prediction values for both DOFs simultaneously. In comparison, two SVR regressors were required to yield the same movement predictions, given that a single SVR regressor can only predict movements for one DOF.

RCNN and SVR movement predictions were then post-processed: (1) they were smoothed using the prediction from the previous window via a moving average filter [79]; (2) predictions between -0.2 and 0.2 were suppressed to 0 [42]; and (3) predictions greater than 1 or less than -1 were clipped to 1 or -1, respectively.

Movement regressors were trained with Trials 4–5 (dynamic) data from a participant and subsequently tested using Trial 6 (dynamic) data from that same participant. Movement regressors were trained, tested, and compared under model specifications S1–S3. The predictive accuracies of these regressors were compared to those of a baseline regressor (BR), trained with only EMG data collected with each participant’s arm at their side (as per standard prosthesis training). For S3, when RCNN movement regression was investigated, RCNN limb position classification was used (that is, S3’s models were all RCNN). Conversely, when SVR movement regression was investigated, LDA limb position classification was used (that is, S3’s models were all statistical). As detailed in the previous Movement Classification section, each movement regressor under S1–S3 and the Baseline was trained with the same three combinations of data streams.

### 3.2.7. Outcome Measures and Statistical Analysis

#### 3.2.7.1. Prediction Accuracy Calculations

**Limb Position Classifiers:** The predicted limb positions performed by the participants were compared to actual limb position classes, with resulting Trial 3 accuracies presented in the Results section as percentages (averaged across participants).

**Movement Classifiers:** The predicted movements performed by the participants were compared to actual movement classes, with resulting Trial 3 accuracies presented in the Results section as percentages (averaged across participants).

**Movement Regressors:** Unlike movement classifiers, which were trained with static data (discrete values), movement regressors were trained and tested with data from dynamic trials (continuous values). As such, the prediction accuracy of movement regressors was determined using  $R^2$  (coefficient of determination) calculations. Recall that the dynamic data consisted of movements that oscillated between either wrist flexion and extension or forearm pronation and supination—that is, in only one DOF at a time. Given this, two kinds of movement periods (or states) occurred for each DOF: active periods, wherein movements were observed in that DOF, and inactive

periods, wherein rest occurred in that DOF while movements were observed in the other DOF, as shown in Figure 3-4.

- For active periods,  $R^2$  values were calculated by comparing movement predictions to the target sinusoids. Resulting Trial 6 values are presented in the Results section as percentages (averaged across participants).
- For inactive periods, however,  $R^2$  values could not be calculated. This is because the actual movements in those periods form a horizontal line at 0 (see Figure 3-4), with  $R^2$  becoming an invalid measure of fit. As such, standard deviations of the movement predictions were calculated instead [42], to reveal the amount of predictive variation. A low standard deviation indicated a high prediction accuracy (that is, one with minimal unwanted movement predictions). Resulting Trial 6 values are presented in the Results section (averaged across participants).

The Kolmogorov-Smirnov test was conducted and revealed that all prediction accuracies did not follow a normal distribution. Therefore, the non-parametric Friedman’s Analysis of Variance and post-hoc Wilcoxon signed-rank tests were used to identify significant prediction accuracy differences across combinations of data streams (for a given classifier or regressor).

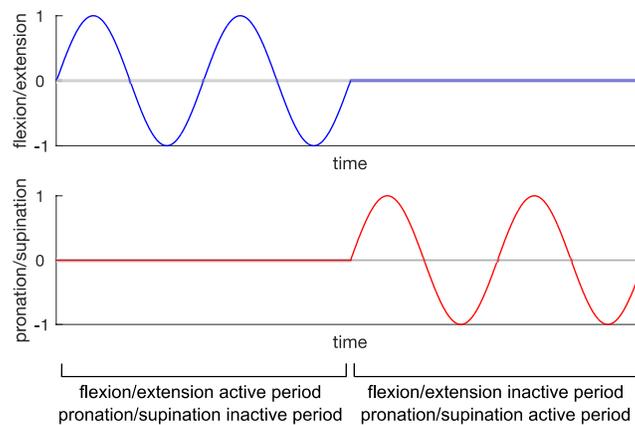


Figure 3-4. Active and inactive periods for each degree of freedom (DOF). An active period is when movements are observed in a given DOF, and an inactive period is when that DOF is in rest while movements are observed in the other DOF.

### 3.2.7.2. Confusion Matrices

Of the movement classifiers and regressors under investigation, the best-performing were further analyzed using confusion matrices. When creating confusion matrices for regressors, predictions and target sinusoid values were categorized into rest, flexion, extension, pronation, and supination using the following rules: (1) values between -0.2 and 0.2 in both DOFs were categorized as rest (in accordance with the chosen post-processing threshold); and (2) for all non-rest values, flexion/extension predictions were compared to pronation/supination predictions, whereby the greater absolute values between them were used to identify movement categories. When creating confusion matrices for the classifiers under further investigation, such categorization was unnecessary.

### 3.2.7.3. Overall Comparisons

Finally, to facilitate direct comparisons between all movement classifiers and regressors under investigation, their root mean square errors (RMSEs) were calculated. RMSE provided a measure of the deviation between predicted and target values. Other studies have used similar measures to compare the performance of classifiers and regressors [44], [52], [80]. To calculate RMSE, movement classification predictions and actual movement classes were converted to values of -1, 0, or 1 in each DOF.

## 3.3. Results

### 3.3.1. Limb Position Classification

The mean limb position classification accuracies (across participants) of the RCNN and LDA classifiers, using four combinations of data streams from both Myo armbands, are shown in Figure 3-5 and Table A-1 (in Appendix A). Notably, both the RCNN and LDA classifiers predicted limb positions most accurately when the IMU's accelerometer data alone were used (99.01% for RCNN, 98.66% for LDA; a 0.35% difference between these).

The mean training and prediction times of the RCNN and LDA classifiers, using the same four combinations of data streams from both Myo armbands, are shown in Table B-1 (in Appendix B). In addition to yielding the highest prediction accuracies, both the RCNN and LDA classifiers resulted in decreased training times when only accelerometer data were used (RCNN: 1.68

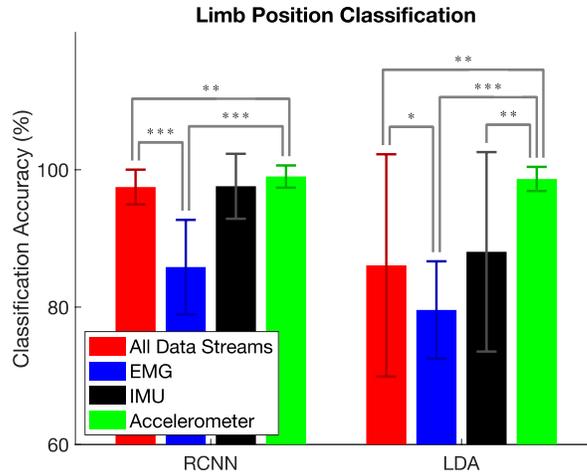


Figure 3-5. Mean limb position classification accuracy (across participants) using RCNN and LDA classification for each combination of data streams: all EMG and IMU data streams from both Myo armbands (red); all EMG data streams from both Myo armbands (blue); all IMU data streams from both Myo armbands (black); and only accelerometer data streams from both Myo armbands (green). One standard deviation of each classification accuracy is shown with error bars, and significant prediction accuracy differences across combinations of data streams are indicated with asterisks (\*:  $p < 0.05$ , \*\*:  $p < 0.01$ , \*\*\*:  $p < 0.001$ ).

minutes, LDA: 38.48 milliseconds) versus when all data streams were used (RCNN: 2.52 minutes, LDA: 89.19 milliseconds). Of note, all classifiers took less than 1 millisecond per prediction, which is well below the 100-millisecond threshold for optimal controller delay [35] (although, admittedly, the computer used in this study was much faster than an embedded processor in a myoelectric prosthesis).

Given these results, for subsequent movement classification and regression investigations, the quaternion and gyroscope data streams from the IMU were eliminated. Furthermore, the limb position classifier in model specification S3 used only accelerometer data streams.

### 3.3.2. Position-Aware Movement Classification: S1–S3

The mean movement classification accuracies (across participants) of the RCNN and LDA classifiers, under each specification and using three combinations of data streams, are shown in Figure 3-6 and Table A-2 (in Appendix A). As expected, the baseline RCNN classifier and baseline LDA classifier yielded the least accurate movement predictions (approximately 85% for each, when using only forearm Myo armband data streams). Overall, the RCNN classifier under S2,

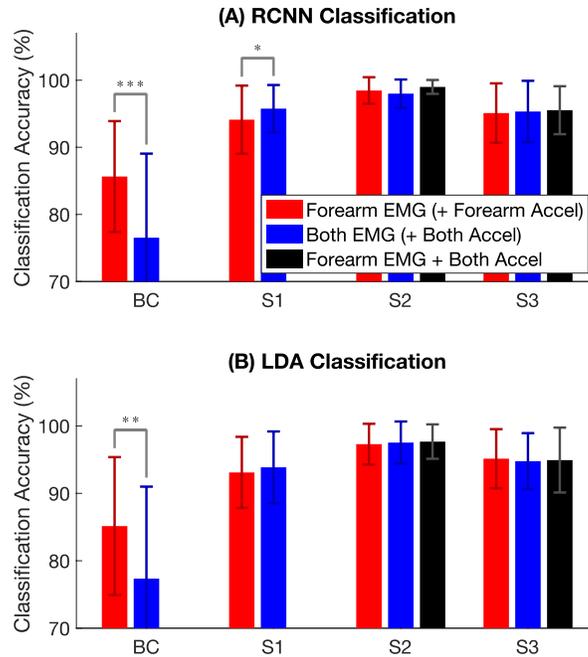


Figure 3-6. Mean movement classification accuracy (across participants) using (A) RCNN classification and (B) LDA classification, under each classification specification: the baseline classifier (BC), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Accuracies are provided for each combination of data streams: data from only the forearm Myo armband (red); data from both Myo armbands (blue); and EMG data from the forearm and accelerometer (Accel) data from both Myo armbands (when applicable, black). One standard deviation of each classification accuracy is shown with error bars, and significant prediction accuracy differences across combinations of data streams are indicated with asterisks (\*:  $p < 0.05$ , \*\*:  $p < 0.01$ , \*\*\*:  $p < 0.001$ ).

trained with EMG data from the forearm Myo armband and accelerometer data from both Myo armbands, yielded the most accurate movement predictions (99.00%). The LDA classifier under S2 using the same training data predicted movements with a slightly lower accuracy (97.67%). So, in comparison, this RCNN classifier was 1.33% more accurate than the corresponding LDA classifier. Generally, most of the position-aware RCNN *and* LDA classifiers yielded movement prediction accuracies over 95%, especially those under S2 and S3.

The mean training and prediction times of the RCNN and LDA classifiers, under each specification and using three combinations of data streams, are shown in Table B-2 (in Appendix B). On average, the RCNN classifiers under S1 and S2 took approximately 2 minutes to train, whereas RCNN classifiers under S3 took 9 minutes to train. The LDA classifiers under S1 and S2 took approximately 26 milliseconds to train, whereas LDA classifiers under S3 took 84 milliseconds to train. When comparing training times across specifications, RCNN and LDA classifiers under S3

required more time than classifiers under other specifications. Of note, all classifiers took less than 6 milliseconds per prediction, which is well below the 100-millisecond threshold for optimal controller delay [35].

Given that the RCNN classifier under S2, trained with EMG data from the forearm Myo armband and accelerometer data from both Myo armbands, predicted movements most accurately, its predictions were further investigated using confusion matrices for each limb position, as shown in Figure 3-7. The RCNN classifier under S2's movement prediction accuracy was found to be consistent across all limb positions, with a roughly equal proportion of errors across classes.

### 3.3.3. Position-Aware Movement Regression: S1–S3

Recall that this study used two outcome measures to assess movement regression predictive accuracy:  $R^2$  values during active periods and standard deviations during inactive periods. The mean  $R^2$  values (across participants) of the RCNN and SVR movement regressors, under each specification and using three combinations of data streams, are shown in Figure 3-8 and Table A-3 (in Appendix A). The corresponding mean standard deviations are presented in Table 3-1. For both flexion/extension and pronation/supination DOFs, the RCNN regressor under S2 yielded the highest  $R^2$  values during active periods and the lowest standard deviations during inactive periods (compared to standard deviations of predictions made with the other RCNN regressors). Overall, the RCNN regressor under S2, trained with EMG and accelerometer data from the forearm Myo armband, yielded high  $R^2$  values for both DOFs (84.93% for flexion/extension and 84.97% for pronation/supination), while reducing the required number of data streams. Conversely, the SVR regressor under S2, also using EMG and accelerometer data from the forearm Myo armband, yielded much lower  $R^2$  values (77.26% for flexion/extension and 60.73% for pronation/supination). The RCNN regressor had  $R^2$  values that were 7.67% greater in flexion/extension and 24.24% greater in pronation/supination than those of the corresponding SVR regressor.

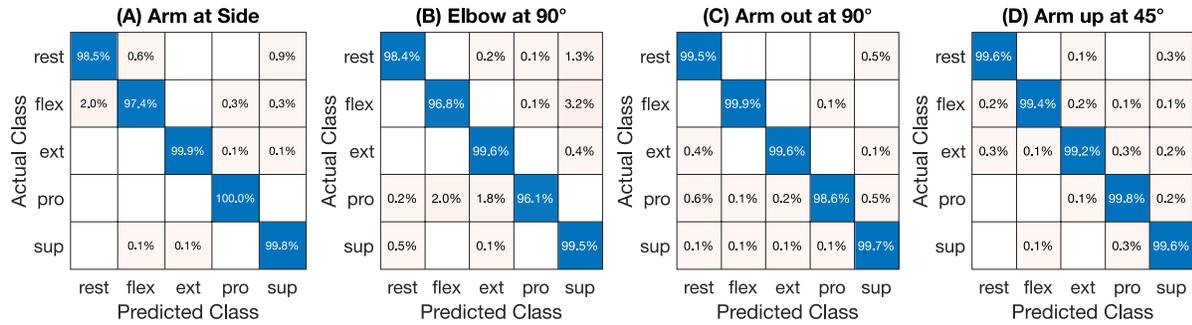


Figure 3-7. RCNN movement classification confusion matrices, across participants for (A) arm at side, (B) elbow at 90°, (C) arm out at 90°, and (D) arm up at 45° under S2 using EMG data from the forearm Myo armband and accelerometer data from both Myo armbands. Movement classes are rest, flexion (flex), extension (ext), pronation (pro), and supination (sup).

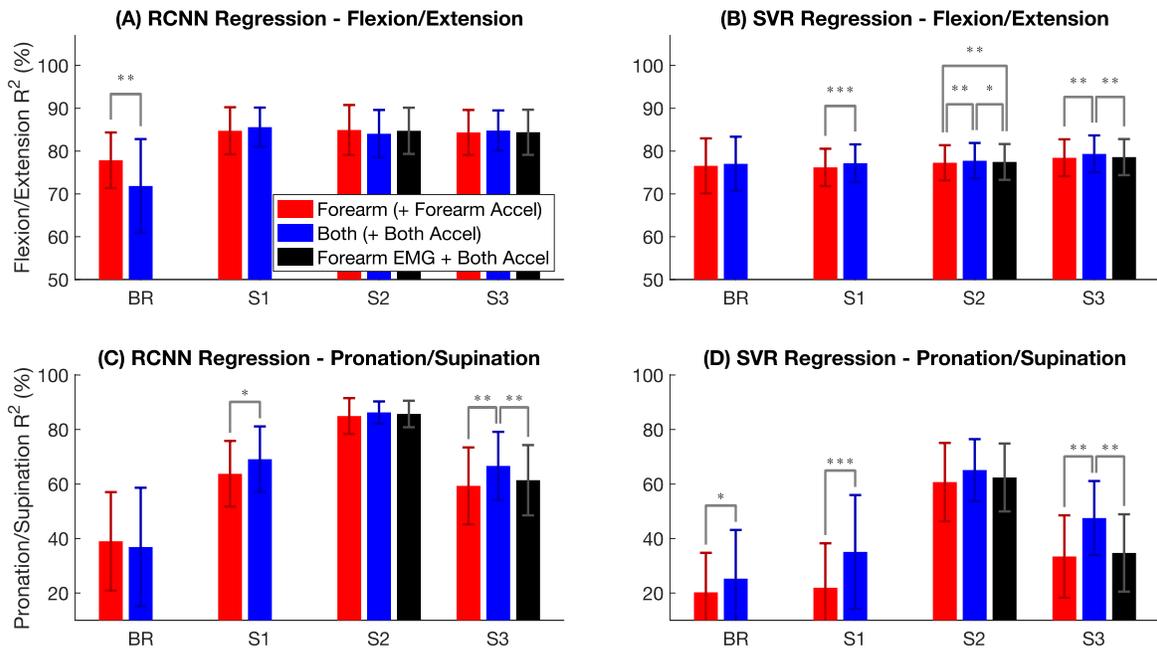


Figure 3-8. Mean movement regression  $R^2$  values (across participants) using (A) RCNN flexion/extension regression, (B) SVR flexion/extension regression, (C) RCNN pronation/supination regression, and (D) SVR pronation/supination regression, under each specification: the baseline regressor (BR), specification 1 (S1), specification 2 (S2), and specification 3 (S3).  $R^2$  values are provided for each combination of data streams: data from only the forearm Myo armband (red); data from both Myo armbands (blue); and EMG data from the forearm and accelerometer (Accel) data from both Myo armbands (when applicable, black). One standard deviation of each  $R^2$  value is shown with error bars, and significant prediction accuracy differences across combinations of data streams are indicated with asterisks (\*:  $p < 0.05$ , \*\*:  $p < 0.01$ , \*\*\*:  $p < 0.001$ ).

When comparing standard deviations, the RCNN regressor under S2 had a flexion/extension standard deviation of 4.20% and a pronation/supination standard deviation of 5.11%. Conversely, the corresponding SVR regressor had standard deviations of 3.19% and 8.10%, for these same movements. The RCNN regressor had a flexion/extension standard deviation 1.01% higher than that of the SVR regressor, and a pronation/supination standard deviation 2.99% lower than that of the SVR regressor.

The mean training and prediction times of the RCNN and SVR regressors under each specification, using three combinations of data streams, are shown in Table B-3 (in Appendix B). On average, the RCNN regressors under S1 and S2 took approximately 1 minute to train, whereas RCNN under S3 took 3 minutes to train. The SVR regressors under S1 and S2 took approximately 21 seconds to train, whereas SVR regressors under S3 took 8 seconds to train. Of note, all regressors took less than 6 milliseconds per prediction, which is well below the 100-millisecond threshold for optimal controller delay [35].

Table 3-1. Regression standard deviations in inactive periods. Mean standard deviations (across participants) using the RCNN flexion/extension regression, SVR flexion/extension regression, RCNN pronation/supination regression, and SVR pronation/supination regression, under each specification: the baseline regressor (BR), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Standard deviations are provided for each combination of data streams: data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands (when applicable).

Specification	Data Streams	Flexion/Extension Standard Deviation (%)		Pronation/Supination Standard Deviation (%)	
		RCNN	SVR	RCNN	SVR
BR	Forearm EMG	10.76	4.85	16.01	13.50
	Both EMG	8.08	5.02	21.15	16.36
S1	Forearm EMG	5.74	3.47	11.14	7.59
	Both EMG	5.89	3.53	8.80	9.73
S2	Forearm EMG + Accelerometer	4.20	3.19	5.11	8.10
	Both EMG + Accelerometer	4.74	3.25	5.76	4.05
	Forearm EMG + Both Accelerometer	4.50	3.13	5.39	4.13
S3	Forearm EMG + Accelerometer	8.81	4.56	11.02	13.71
	Both EMG + Accelerometer	6.56	4.97	8.91	13.82
	Forearm EMG + Both Accelerometer	7.70	4.23	11.02	13.21

Given that the RCNN regressor under S2, trained with EMG and accelerometer data from the forearm Myo armband, predicted movements most accurately, its predictions were further investigated. These predictions were categorized into movement classes (rest, flexion, extension, pronation, and supination), and the resulting confusion matrices for each limb position were generated, as shown in Figure 3-9. The RCNN regressor under S2’s movement prediction accuracy was found to be consistent across all limb positions, but most errors were related to rest.

### 3.3.4. Results Summary

A comparative summary of the classifiers and regressors that were investigated in this study is presented in Table 3-2, wherein the RMSE was calculated for all movement predictions. Table 3-2 identifies the position-aware control strategies that most accurately predicted movements — those with an RMSE less than 0.22 (this threshold was chosen as it represents the 70<sup>th</sup> percentile of accuracy).

Overall, the best classifier was determined to be the RCNN classifier under S2, and the best regressor was the RCNN regressor under S2—both yielded the most accurate movement predictions, while using fewer than all available data streams.

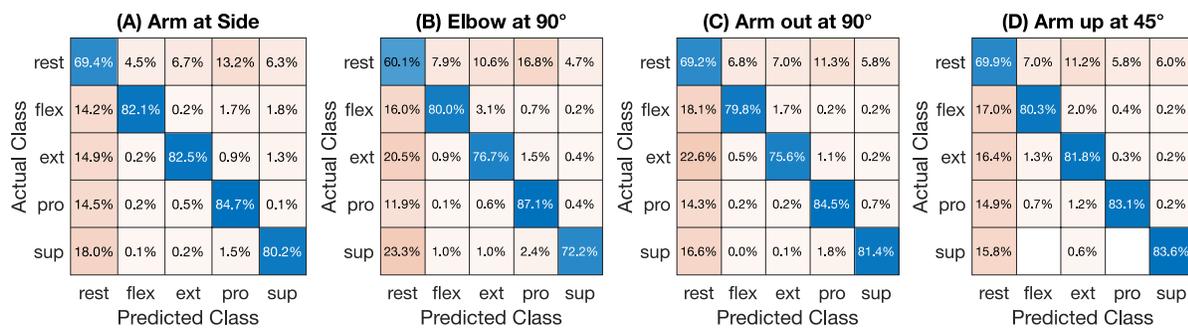


Figure 3-9. RCNN movement regression confusion matrices, across participants for (A) arm at side, (B) elbow at 90°, (C) arm out at 90°, and (D) arm up at 45° under S2 using data from the forearm Myo armband. Movement classes are rest, flexion (flex), extension (ext), pronation(pro), and supination (sup).

Table 3-2. Movement prediction accuracy summary. Root mean square error across participants for each movement prediction method (classification or regression), specification (BC, BR, S1, S2, and S3), type of model (RCNN, LDA, and SVR), and combination of data streams (data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands, when applicable). Root mean square errors less than 0.22 are highlighted in green.

Model and Specification			Root Mean Square Error (calculated with different data streams)		
			Forearm EMG + Accelerometer	Both EMG + Accelerometer	Forearm EMG + Both Accelerometer
Classifier	BC	RCNN	0.36	0.49	
		LDA	0.40	0.50	
	S1	RCNN	0.24	0.20	
		LDA	0.27	0.24	
	S2	RCNN	0.11	0.14	0.09
		LDA	0.16	0.15	0.15
	S3	RCNN	0.23	0.22	0.22
		LDA	0.24	0.23	0.24
Regressor	BR	RCNN	0.34	0.36	
		SVR	0.37	0.36	
	S1	RCNN	0.26	0.24	
		SVR	0.36	0.34	
	S2	RCNN	0.20	0.20	0.20
		SVR	0.28	0.27	0.28
	S3	RCNN	0.27	0.25	0.27
		SVR	0.34	0.31	0.34

### 3.4. Discussion

The goal of this study was to investigate RCNN-based position-aware myoelectric prosthesis control strategies, using combined EMG and IMU input data streams. EMG signals primarily informed intended movements, whereas IMU signals primarily provided context about limb position. Classifiers and regressors used these signals to make position-aware movement predictions. Recall that three model specifications were explored:

S1 – Model (classifier or regressor) trained with EMG data from all limb positions

S2 – Model trained with EMG and IMU data from all limb positions

S3 – Models trained with EMG data at each limb position, with subsequent predictions occurring in a 2-staged sequence: 1<sup>st</sup>, a limb position was classified using IMU data; 2<sup>nd</sup>, a corresponding model (trained at that specified limb position) predicted a wrist movement using EMG data

For this study, a favourable position-aware myoelectric prosthesis control strategy was considered to be one where a classifier or regressor yielded accurate movement predictions, using the fewest possible data streams.

### 3.4.1. Position-Aware Classification

This study corroborates and extends the findings of earlier prosthesis control strategy research that likewise used classifiers under model specifications S1–S3. Such research yielded improved movement predictions compared to a baseline classifier [6], [12], [81]. Fougner et al. found that LDA classification under S2 yielded the most accurate movement predictions [15], whereas Geng et al. concluded that LDA classification under S3 proved to be the most accurate [22], [23].

Of the position-aware classifiers under S1–S3 that were investigated in this study, **the most promising was the RCNN classifier under S2 (with EMG data from the forearm Myo armband and accelerometer data from both Myo armbands)**. It yielded the highest movement prediction accuracy (99.00%, versus the LDA's at 97.67%) while requiring a reduced number of data streams. The success of this classifier under S2 is consistent with Fougner et al.'s observations [15]. Notably, classifiers under S1 performed less accurately compared to those under S2 because accelerometer data (and consequently limb position information) was not included under S1. Additionally, classifiers under S3 performed less accurately than those under S2, likely because the classification sequence of S3 (with two stages) introduced the potential to compound errors.

This study's RCNN classifier under S2 yielded more accurate movement predictions than did classifiers in earlier research. As such, this work offers encouraging results towards solving the limb position effect.

### 3.4.2. Position-Aware Regression

To our knowledge, only one other study has implemented a regression-based device control strategy in the context of addressing the limb position effect. Park et al. employed a position

decoder to accomplish position-independent regression, and tested their resulting predictive device control outcomes through real-time experimentation [61]. They predicted movements with smaller  $R^2$  values than those of this study, but caution should be taken when comparing their real-time results to those of this offline work.

Of the position-aware regressors under S1–S3 that were investigated in this study, **the most promising was the RCNN regressor under S2 (with EMG and accelerometer data from the forearm Myo armband)**. It yielded the highest movement prediction accuracy (with  $R^2$  values of 84.93% for wrist flexion/extension and 84.97% for forearm pronation/supination, versus the SVR's at 77.26% and 60.73%, respectively). It also required a reduced number of data streams. However, this RCNN regressor predicted movements with lower accuracies than the investigated classifiers. This is in keeping with previous research that found regression to be less accurate than classification, due to the increased complexity of regression predictions (continuous values for each DOF) [30]. Despite being lower in predictive accuracy than classification, regression may offer increased functionality, through both simultaneous and proportional control, and as such might outperform classification in real-time experimentation [80].

Of the errors that contributed to the decreased accuracy of the RCNN regressor under S2, the majority occurred around the rest periods, as evidenced in Figure 3-9. These errors can be categorized as either false negatives (falsely predicting rest) or false positives (falsely predicting a movement instead of rest). False negatives occurred more frequently. Notably, false negatives can be considered acceptable in prosthesis control; that is, simply perceived as device responsiveness latency by users [82]. The detected false negatives may have resulted from prediction suppression, whereby prediction values between -0.2 and 0.2 were set to 0. Note that in future work, this suppression threshold can be adjusted. Finally, both false negatives and false positives may have been caused by offsets between the participants' movements and the sinusoids chosen to represent these movements. Undoubtedly, without a perfect match between sinusoids and movements, slight inaccuracies can be expected.

To mitigate the occurrence of such inaccuracies, participants' movements must closely track training sinusoids. It is a common research practice to have participants follow an onscreen training target (such as a moving cursor or virtual hand) [42], [83], [84], [85]. But this practice can result in the introduction of participant movement delays. This study corrected the delay between

onscreen movement instructions and participants' actual movements by using generated sinusoids for both wrist flexion/extension and forearm pronation/supination. To accomplish this, peak muscle contractions were extracted from the EMG signal data and used to produce wrist flexion/extension sinusoids, whereas accelerometer signals were used to generate forearm pronation/supination sinusoids. Despite making the necessary movement corrections in this study through the use of sinusoids, offsets may have still been present (although presumably smaller than without such corrections).

To further reduce the occurrence of movement offsets, modifications could be made to the data collection methods for the regression training routine. For example, if a participant were to follow an onscreen sinusoid overlaid with their real-time EMG signals [86] (that is, afforded visual feedback), more accurate instruction adherence would likely result. That same sinusoid could then be used as a precise training target (as opposed to extracting muscle and position signals' peaks for sinusoids). Additionally, if participants were required to complete a practice dynamic trial before data collection, the precision with which they follow the target sinusoid would likely improve.

### 3.4.3. Promising RCNN Outcomes

As expected, the RCNN-based control strategies investigated in this study predicted movements more accurately than statistical-based alternatives (which was especially evident when comparing RCNN and SVR regressors). This may be because RCNNs offer the advantage of learning new features from complex input data. Other studies have investigated the use of engineered feature sets to address the limb position problem, and as such did not harness this advantage [60], [62], [65]. Despite yielding position-aware movement predictions using engineered features, their models did not perform quite as well as this study's RCNN classifier under S2. Although these studies examined more extensive limb position ranges and movements, their lower predictive accuracies may suggest that for position-aware myoelectric control, *learning* new features with RCNNs may be favourable over using *engineered* features. Naturally, further research is required to confirm this.

#### 3.4.4. Limitations

Limitations in this study included: the requirement for training routines with long durations; the number of limb positions and wrist movements used for training and testing models was not exhaustive; models were only tested on the 3<sup>rd</sup> or 6<sup>th</sup> trials; more training data may be required for accurate results in other limb positions; and only static limb positions were employed in this study (training with continuous limb positions may improve predictive accuracy [19], [21]).

Notably, regressors were tested using data from oscillations in one DOF at a time. This testing method does not demonstrate model performance during simultaneous muscle contractions in two DOFs and consequently cannot translate directly to activities of daily living. Furthermore, data from only *isotonic* muscle contractions were recorded, rather than data resulting from isometric contractions (which are used to control a prosthesis). The performance of the models presented in this study may differ when isometric contractions are used.

Finally, the feasibility of implementing RCNN-based prosthesis control strategies using existing hardware was not investigated in this study. However, as the capabilities of onboard prosthetic device processors continue to improve, it is expected that implementation might well be possible in the near future.

#### 3.4.5. Future Work

Future work will focus on real-time testing of the promising RCNN-based control strategies (RCNN classifier and regressor under S2) presented in this study. Upcoming research will include real-time testing of these control strategies with both non-disabled participants (using a simulated prosthesis) and myoelectric prosthesis users. Testing using a simulated or actual prosthesis will require participants to use isometric contractions for device control. Participants will carry out functional tasks that simulate activities of daily living. These tasks will also allow for the assessment of regression control for simultaneous movements.

Although the movement prediction accuracy of myoelectric control strategies may not always correlate with their real-time performance [23], [66], a reduction in the limb position effect can be expected in real-time experimentation (given that participants will have visual feedback and will be able to adjust their muscle contractions accordingly [79]). Improvements to the regression

training routine that were gleaned from this study will be implemented in future work. Additionally, as RCNN classifiers under S2 required training routines with long durations (relative to the baseline classifiers), a generalized RCNN classifier will be investigated, with the goal of eliminating the training routine (and consequently model training time) altogether.

### 3.5. Conclusion

This study has identified two promising position-aware myoelectric prosthesis control strategies towards solving the *limb position effect* problem:

- (1) An RCNN classifier trained with EMG and accelerometer (IMU) data (captured from participants across multiple limb positions) predicted movements best, while requiring a reduced number of data streams; and
- (2) An RCNN regressor trained with EMG and accelerometer data (captured from participants across multiple limb positions) performed much better than an SVR regressor, although not as accurately as the aforementioned RCNN classifier. It also required fewer than all available data streams.

It is expected that both of these RCNN-based control strategies will likewise yield accurate, position-aware movement predictions in real-time experimentation. As such, results of this research are anticipated to improve the usability of myoelectric devices for individuals with amputation, particularly when faced with the challenges of the *limb position effect*.

# Chapter 4.

## Composite RCNN Classification Control

Chapter 4 presents “Composite Recurrent Convolutional Neural Networks Offer a Position-Aware Prosthesis Control Alternative While Balancing Predictive Accuracy with Training Burden,” originally published in the Proceedings of the International Consortium on Rehabilitation Robotics, in 2022 [11]. It investigates a novel control strategy (hereinafter referred to as a control solution) that addresses burdensome training routines identified in Chapter 3. In this current chapter, a *composite* recurrent convolutional neural network is developed and tested. The goal of this work was to minimize the need for model retaining/calibration during daily device use, thereby reducing overall training burden. An illustration of how the information presented in Chapter 4 contributes to the overall work in this thesis is presented in Figure 4-1.

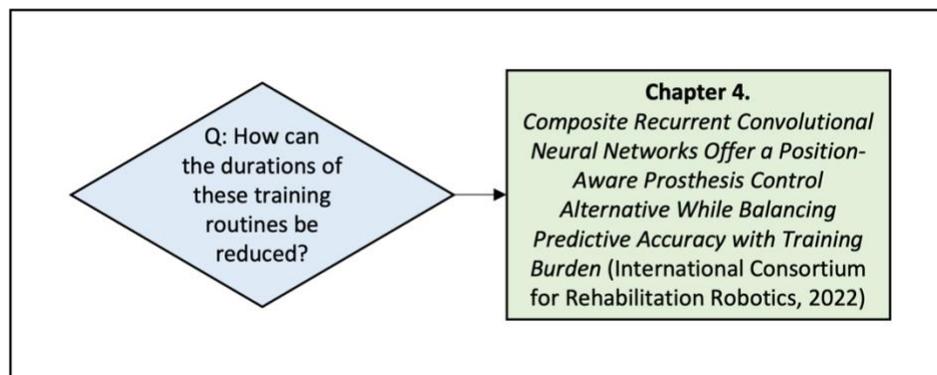


Figure 4-1. Research question addressed by Chapter 4.

## Abstract

To mitigate the *limb position effect* that hinders myoelectric upper limb prosthesis control, pattern recognition-based models must accurately predict user-intended movements across a multitude of limb positions. Such models can use electromyography (EMG) and inertial measurement units to capture necessary multi-position data. However, this data capture solution requires lengthy user-performed model training routines, with movements in many limb positions, plus retraining thereafter due to inherent signal variations over time. While a general-purpose control model (trained with a dataset that represents numerous device users) eliminates the user-training requirement altogether, it yields low movement predictive accuracy. Conversely, a user-specific control model (trained with a smaller dataset from an individual) yields high predictive accuracy, but requires retraining over time. This study capitalizes on the benefits offered by both such control options, and contributes an alternative control solution—a novel recurrent convolutional neural network (RCNN)-based *Composite Model* that combines the *representation* portion of a general-purpose model, with the *decision* portion of a user-specific model. The resulting Composite Model offers moderate movement predictive accuracy across various limb positions and a reduction in user training routine requirements, suggesting a new research direction to help mitigate the limb position effect along with model training burden.

### 4.1. Introduction

Pattern recognition-based solutions for upper limb myoelectric prosthesis control use muscle activations from a user’s residual limb to predict user-intended movements [13]. These solutions are implemented via a control model that learns prediction rules based on muscle signal feature patterns that are gleaned from movement data—be it from offline pre-recorded data and/or online real-time data. This learning process is known as model training. Electromyography (EMG) surface electrodes embedded in a device socket can capture necessary movement data [6], whereas inertial measurement units (IMUs) can be used to capture complementary positional data [10]. Together, EMG and IMU data can inform a control model that functions across various upper limb positions [10], which is particularly important when addressing the challenge of the *limb position effect* [18]. This effect is a well-documented consequence of EMG signal alteration that occurs when a user attempts to use their device in a position different from that in which its pattern

recognition-based control model was initially trained [14], [18]. The limb position effect is known to cause degradation of device control, unexpected prosthesis movements, and user frustration [18].

Training a pattern recognition-based model with both EMG and IMU data from multiple limb positions can effectively mitigate the limb position effect [15], [22]. However, a major drawback to this control solution is that device users must execute a long training routine before device use (to capture movement data from a broad range of limb positions). Such a training routine requires that wrist and hand movements be executed in positions ranging from arm at side to above head [10]. Another drawback is that pattern recognition-based control normally requires *retraining*, or the execution of a training routine again [87], [88], [89], in response to muscle fatigue and expected electrode shifts [14]. Repeated execution of a long training routine becomes quite burdensome for a prosthesis user—a negative consequence of offering device control across various limb positions [10].

To improve position-aware pattern recognition-based device control, some prosthesis researchers have turned to deep learning methods, including recurrent convolutional neural networks (RCNNs) [10], [28], [72]. RCNNs can be advantageous over traditional linear discriminant analysis (LDA) [10]. RCNNs offer the ability to: (1) learn new features from raw EMG and IMU signals (that is, to represent input signals), (2) combine high volumes of data from multiple sensors, and (3) recognize the nuances of the time-varying behaviour of EMG signals. RCNN-based control models can harness these abilities to classify user-intended limb movements and recognize movement patterns. With such control, RCNNs store the details learned through model training as weights (numbers which are initially random, but later are established through training). These weights are stored in layers, which together form an RCNN architecture. The general architecture of an RCNN-based model includes: an input layer (that receives input signals); a *representation* portion (where information about how to represent the input signals is stored); a *decision* portion (where time-related signal behaviours are taken into consideration, all details about the signals are brought together, and movement class decisions are made); and an output layer (that relays predicted movement instructions to the device motors).

With RCNN-based model architecture understood, the challenge of how to mitigate the limb position effect with this type of model poses a prosthesis control solution trade-off. That is, would

it be more prudent to develop either: (A) a general-purpose control model, trained with a large dataset from a variety of device users—which expectedly would be robust to changes in EMG signals, but at the expense of a low user-intended movement predictive accuracy, or (B) a user-specific model, trained with a smaller set of their personal data—which expectedly would have a high predictive accuracy, but would need to be retrained as conditions change over time? Consider that there might be a third option, wherein the representation portion of an RCNN-based model could be trained with data from numerous people (to learn robust model features), and the decision portion could be trained with data from one specific person (to tailor the model’s predictions to this individual). Such a combined upper limb device control solution might strike a user-oriented balance between movement predictive accuracy and training routine burden.

This study verifies that a well-balanced control approach can be achieved by uniquely combining layers from a general-purpose RCNN-based model with those of user-specific RCNN-based models—as a means of addressing the impact of the limb position effect on myoelectric prosthesis control, while easing burdensome model training for users. Given that together, EMG and IMU (specifically, accelerometer) data can be used for position-aware control [10], both were used in the combined, or *composite*, control solution. Three RCNN-based model types were developed, using offline data collected from participants who performed specific movements in various limb positions and under time-delayed conditions: (1) a general-purpose control model (**Generalized Model**) offered a “one size fits all” solution, (2) personalized control models (**Personalized Models**) offered “custom to user” solutions, and (3) novel composite control models (**Composite Models**) uniquely combined the representation portion of the Generalized Model architecture with the decision portions of Personalized Model architectures. The Composite Models were expected to predict movements with a mid-range accuracy (between those of the Generalized Model and the Personalized Models), across all time-delayed conditions. This study contributes to the literature by proposing an alternative deep learning-based myoelectric device control solution—one that uniquely combines “best of” RCNN model architecture portions.

## 4.2. Methods

The same experimental methods were used to collect a large general-purpose dataset along with individual pilot study datasets, each of which were captured as participants performed specific movements in four limb positions. The pilot study participants had time-delays introduced

during their data collection trials, to simulate muscle signal variations over time. Model testing was conducted under initial versus time-delayed conditions. The predictive accuracies of the resulting models were subsequently analyzed.

#### 4.2.1. Participants

**General Participant Group:** A total of 19 participants with no upper-body pathology or recent neurological or musculoskeletal injuries were recruited. 3 participants had previous experience with EMG control, all had normal or corrected to normal vision, 17 were right-handed, 10 were male, and the participants had a mean age of  $26.4 \pm 8.1$  years ( $\pm 1$  standard deviation).

**Pilot Study Participant Group:** 5 new participants (who were not part of the general group) were recruited. Each was without upper-body pathology or recent neurological or musculoskeletal injuries. 2 participants had previous experience with EMG control, all had normal or corrected to normal vision, all were right-handed, 3 were male, and the participants had a mean age of  $23.0 \pm 4.8$  years.

The experiment protocol was approved by the University of Alberta Health Research Ethics Board (Pro00086557). Each participant provided written informed consent.

#### 4.2.2. Experimental Setup

A Myo gesture control armband (Thalmic Labs, Kitchener, Canada) was used to collect EMG and IMU data. The armband was equipped with 8 surface electrodes and an IMU. EMG data were collected by the electrodes (sampled at 200 Hz) and arm position data were collected by the IMU (sampled at 50 Hz). Using Myo Connect software, the EMG and IMU data were streamed into Matlab running on a laptop. A custom Matlab script captured the data and simultaneously displayed onscreen instructions for a participant to follow.

#### 4.2.3. Data Collection

Participants in the general and pilot study groups donned a Myo armband on their self-identified dominant arm, as shown in Figure 4-2A. The armband was worn at approximately the upper third of their forearm (an average of  $37.3 \pm 1.6\%$  of the way from the lateral epicondyle to the radial styloid).

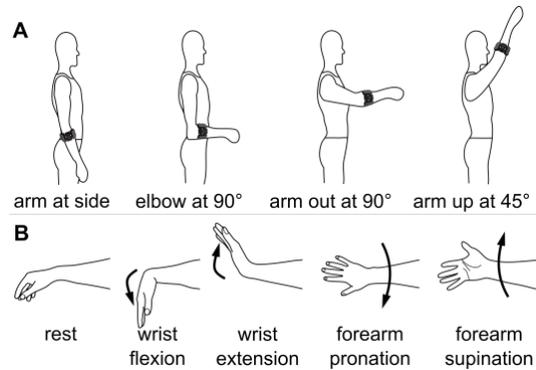


Figure 4-2. (A) The placement of each Myo armband, along with limb positions used in data collection, and (B) movements used in data collection, adapted from Williams et al. [10] with permission.

All participants followed onscreen instructions, performing 5 second static muscle contractions that resulted in rest, wrist flexion, wrist extension, forearm pronation, and forearm supination movements, as shown in Figure 4-2B. Participants were instructed to perform contractions at a moderate effort that could be sustained for 5 seconds. Short rest periods were provided to participants after each contraction. All contractions were repeated in 4 arm positions: arm at side, elbow bent at 90°, arm out in front at 90°, and arm up at 45° from vertical, as shown in Figure 4-2A.

**General Participant Group Trials:** Each general participant completed 3 data collection trials. The resulting data formed a large *General Dataset* of 57 trials (3 trials x 19 participants).

**Pilot Study (PS) Participant Group Trials:** Each pilot study participant completed 9 data collection trials (*Trials PS1–PS9*), which were divided into 3 data blocks.

*Block 1:* The first block of 3 trials (PS1–PS3) were conducted after the Myo armband was initially donned (initial use condition).

*Block 2:* The second block of 3 trials (PS4–PS6) were conducted after a 15-minute break had elapsed (first time-delayed condition), during which the participants doffed the Myo armband, used their arm for non-testing purposes, and then again donned the Myo armband.

*Block 3:* The final block of 3 trials (PS7–PS9) were conducted after another 15-minute break (second time-delayed condition), during which the participants doffed and donned the Myo armband again, as per Block 2.

#### 4.2.4. Data Pre-Processing

The EMG data from the Myo armband were filtered using a high pass filter at 20 Hz, as well as a notch filter at 60 Hz to remove electrical noise. Then, the accelerometer (IMU) data streams were resampled to 200 Hz to align them with the corresponding EMG data. The trials were segmented into individual muscle contractions (rest, wrist flexion, wrist extension, forearm pronation, and forearm supination). Then, the data from all trials were segmented further into 160 ms windows for RCNN-based model implementation.

#### 4.2.5. RCNN-Based Model Architecture

Visual representations of the RCNN-based Generalized, Personalized, and Composite Model architectures are provided in Figure 4-3A, B, and C–G, respectively. The control models each consisted of 27 layers. The Generalized Model layers included all convolution layers and formed the representation portion of the Composite Model, from which EMG and accelerometer data features were extracted. The Personalized Model layers formed the decision portion of the Composite Model, wherein movements were classified based on learned EMG and accelerometer data features. The model architecture was developed using Matlab 2021a software, with the number of filters, the filter size, the pooling size, and the number of epochs determined through Bayesian optimization. All model development was performed using an Intel® Core™ i9-9900K CPU (3.60 GHz).

##### 4.2.5.1. Model Hyperparameters

In this study, hyperparameters that were uniquely suitable to each model were selected. This ensured that each model was afforded the best opportunity for high predictive accuracy.

**Generalized Model:** Hyperparameters suitable to *all* general participants were selected. Bayesian optimization was performed across the entire General Dataset in 2 phases: Phase 1 used a wide range for each of the hyperparameters that were being optimized; and Phase 2 used a narrower range of values centered at the optimal found in Phase 1. The resulting values are shown in Figure 4-3A.

**Personalized Models:** Hyperparameters suitable to *all* general participants were selected, as per the Generalized Model optimization methods. It would be infeasible to optimize hyperparameters

for each new set of training data, given that the Personalized Models were expected to require frequent retraining. The resulting values are shown in Figure 4-3B.

**Composite Models:** Hyperparameters that were *unique to each* pilot study participant, as presented in Figure 4-3C–G, were selected using the Generalized Model optimization methods (performed for each pilot study participant). Unlike for the Personalized Models, it *would* be feasible to optimize hyperparameters for each new set of training data for Composite Models. This is because the Composite Models would require infrequent retraining in comparison, thereby leaving time available for optimization. As such, hyperparameters were optimized for each pilot study participant—affording each Composite Model the best opportunity to achieve high movement predictive accuracy.

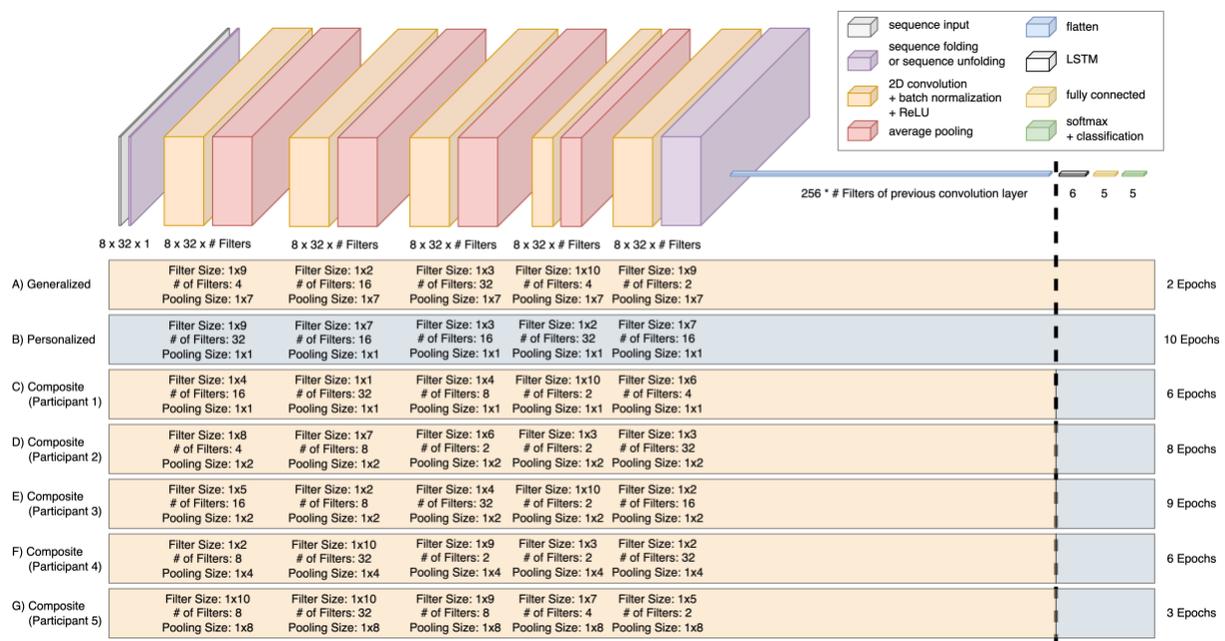


Figure 4-3. Generalized (A), Personalized (B), and Composite (C–G) RCNN-based myoelectric prosthesis control models—each composed of 27 layers. Layers included: sequence input layer; sequence folding layer; 4 groups of convolution, batch normalization, rectified linear unit (ReLU), and average pooling; 1 group of convolution, batch normalization, and ReLU; sequence unfolding layer; flatten layer; long short-term memory (LSTM) layer; fully connected layer; softmax layer; and classification layer. The dashed line separates the layers of the Generalized and Personalized Models that were used to form the Composite Models. Optimized convolution filter sizes, number of filters, pooling sizes, and number of epochs are shown for each model.

## 4.2.6. Model Training & Testing

In this study, 1 Generalized Model was developed. In addition, for each of the 5 pilot study participants, 3 Personalized Models and 1 Composite Model were developed. All resulting models yielded a predicted movement class (rest, wrist flexion, wrist extension, forearm pronation, and forearm supination) for each window. These movement classes were learned (that is, associations between EMG signal data and movements were established) during RCNN-based Matlab model training using offline data.

### 4.2.6.1. Model Training Process

**Generalized Model:** The Generalized Model was trained using the movement classes and data from the entire General Dataset—to establish a once-generally-trained model.

**Personalized Models:** 3 Personalized Models were custom-trained for each pilot study participant, using movement classes and data resulting from only the first 2 trials from each of their 3 data blocks (note that the third such trials were reserved for model testing). These 3 models were structurally the same, but trained with different data for each participant:

- *Block 1 Personalized Model* was trained with the initial use condition data from Trials PS1 and PS2
- *Block 2 Personalized Model* was trained with the first time-delayed condition data from Trials PS4 and PS5
- *Block 3 Personalized Model* was trained with the second time-delayed condition data from Trials PS7 and PS8

Time-delayed conditions were introduced to simulate control model degradation over time (when model retraining might be required by a prosthesis user [87], [88]). Overall, this time-delayed training process established “retrained” Personalized Models for each pilot study participant.

**Composite Models:** 1 Composite Model was developed for each pilot study participant and combined the first layers of the Generalized Model, with the last layers of *only* their Block 1 Personalized Model. This process established a once-trained Composite Model for comparative analysis purposes. The use of only the Block 1 Personalized Model is intended to mimic actual

prosthesis control, wherein a model is first trained and then employed in real-time afterwards (that is, with Block 1 data captured prior to that of Blocks 2 and 3).

#### 4.2.6.2. Model Testing Process

The testing process compared the pilot study participants' predicted movements to their actual movements and yielded predictive accuracies for all models under investigation. Testing was conducted using each pilot study participant's third data trial from each of their 3 data blocks (Trials PS3, PS6, and PS9).

**Generalized Model:** Trial PS3, PS6, and PS9 data were used to test the Generalized Model. Consistently low predictive accuracies were anticipated, given that the Generalized Model was not participant-specific in nature.

**Personalized Models:** Each pilot study participant's Personalized Models were tested under the following scenarios to confirm that device model retraining would indeed improve user control [87], [88].

*With retraining:* Each Block 1 Personalized Model was tested using Trial PS3 (initial use condition) data. Each Block 2 Personalized Model and Block 3 Personalized Model were similarly tested using Trial PS6 (first time-delayed condition) and Trial PS9 (second time-delayed condition) data, respectively. High predictive accuracies under each condition were anticipated, given the retrained nature of these models.

*Without retraining:* Each Block 1 Personalized Model was also tested using Trial PS6 (first time-delayed condition) and Trial PS9 (second time-delayed condition) data. Given that new training data were not used to retrain Block 2 and 3 Personalized Models, diminished predictive accuracies under these testing conditions were anticipated.

**Composite Models:** Trial PS3, PS6, and PS9 data were used to test the Composite Models. Predictive accuracies that were consistently higher than those of the Generalized Model and lower than those of the retrained Personalized Models were anticipated.

## 4.3. Results

### 4.3.1. Generalized Model Performance

The Generalized Model, as tested across all pilot study participants' Trial PS3, PS6, and PS9 data, yielded mean ( $\pm 1$  standard deviation) predicted movement accuracies of  $75.55 \pm 3.44\%$ ,  $81.66 \pm 3.05\%$ , and  $77.11 \pm 3.11\%$ , respectively (Figure 4-4A, blue). As the Generalized Model did not require training by the pilot study participants (the General Dataset was used), the cumulative training routine duration remained at 0 minutes (Figure 4-4A, red).

### 4.3.2. Personalized model performance

*With retraining:* The retrained Personalized Models, as tested across all pilot study participants' Trial PS3, PS6, and PS9 data, yielded mean ( $\pm 1$  standard deviation) predicted movement accuracies of  $97.06 \pm 1.23\%$ ,  $97.97 \pm 0.72\%$ ,  $98.13 \pm 0.59\%$ , respectively (Figure 4-4B, blue). The cumulative training routine duration for the retrained Personalized Models steadily increased, given that model retraining under each time-delayed condition required new data (Figure 4-4B, red).

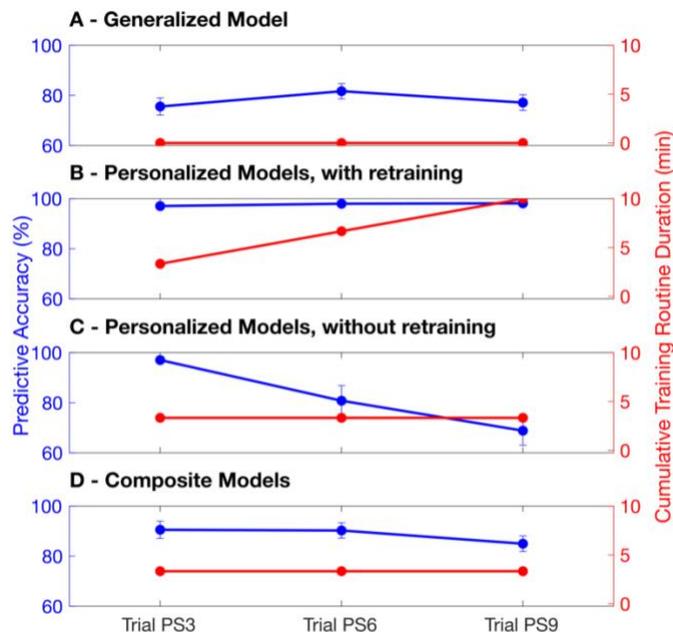


Figure 4-4. Mean predictive accuracy (blue) and cumulative training routine durations (red) for the (A) Generalized Model, (B) Personalized Models with retraining, (C) Block 1 Personalized Models without retraining, and (D) Composite Models, when tested across all pilot study participants' Trial PS3, PS6, and PS9 data.

*Without retraining:* The Block 1 Personalized Models, as tested across all pilot study participants' Trial PS3, PS6, and PS9 data, yielded mean ( $\pm 1$  standard deviation) predicted movement accuracies of  $97.06 \pm 1.23\%$ ,  $80.81 \pm 6.05\%$  and  $68.85 \pm 5.81\%$ , respectively (Figure 4-4C, blue). The cumulative training routine duration for the Block 1 Personalized Models did not increase after the initial use condition, as no retraining took place thereafter (Figure 4-4C, red).

### 4.3.3. Composite Model Performance

The Composite Models, as tested across all pilot study participants' Trial PS3, PS6, and PS9 data, yielded mean ( $\pm 1$  standard deviation) predicted movement accuracies of  $90.53 \pm 1.36\%$ ,  $90.24 \pm 2.16\%$ , and  $84.96 \pm 2.30\%$ , respectively (Figure 4-4D, blue). Interestingly, the predictive accuracies of the Composite Models under time-delayed conditions were higher than those of the Block 1 Personalized Models—testing with Trial PS6 data yielded 9.44% higher accuracy and testing with Trial PS9 data yielded 16.11% higher accuracy. The cumulative training routine duration for the Composite Models did not increase after the initial use condition, as no retraining took place thereafter (Figure 4-4D, red).

### 4.3.4. Results Summary

Figure 4-5 illustrates each model's mean predictive accuracy across all pilot study participants as well as across Trial PS6 and PS9 data, in order of increasing predictive accuracy.

**Personalized Models without retraining:**  $74.83 \pm 9.07\%$

**Generalized Model:**  $79.38 \pm 4.66\%$

**Composite Models:**  $87.60 \pm 3.44\%$

**Personalized Models with retraining:**  $98.05 \pm 0.98\%$

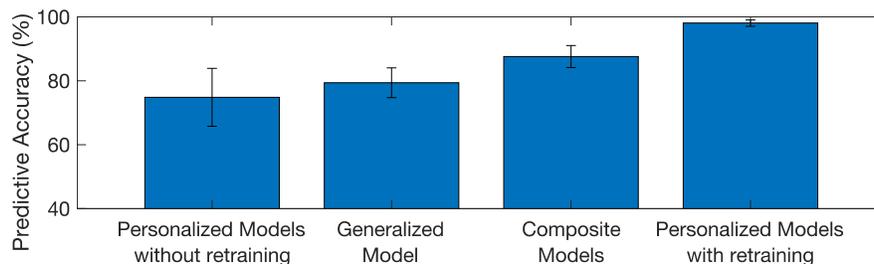


Figure 4-5. Mean predictive accuracy of the Personalized Models without retraining, Generalized Model, Composite Models, and Personalized Models with retraining, under time-delayed conditions when tested across all pilot study participants as well as across Trial PS6 and PS9 data. Error bars indicate  $\pm 1$  standard error of each mean.

#### 4.3.5. Overfitting assessment

To assess overfitting, the predictive accuracies of the Generalized, Personalized and Composite models were analyzed—their respective *training* dataset accuracies were compared to those resulting from their *testing* data as reported in this study. The Generalized Model achieved an accuracy of 95.18% when tested using the General Dataset. Although this value is 15.80% higher than the accuracy reported herein (and as such, could indicate overfitting), the model was only trained for 2 epochs, rendering it unlikely to be overfitted. Conversely, the Personalized Models achieved an average accuracy of  $99.99\% \pm 0.02\%$  when tested using their training and retraining data, which is only 1.94% higher than the accuracy reported herein. As this accuracy difference is small, there likely was no overfitting. Finally, the Composite Models achieved an average accuracy of  $92.73 \pm 6.46\%$  when tested using the training data from each pilot study participant, which is only 5.13% higher than the accuracy reported herein. This small accuracy difference also suggests no overfitting.

### 4.4. Discussion

This study proposes a novel RCNN-based Composite Model concept for myoelectric prosthesis control towards addressing the challenge of the limb position effect, with the goal of reducing the training routine burden placed on users. This work confirmed that a Generalized Model performs as a “one-size fits all” control solution—offering low movement predictive accuracy, but with the benefit of requiring no user training. It also confirmed that Personalized Models offer user-specific control that can be retrained as required, thereby performing as “custom fit” control solutions—offering high movement predictive accuracy across various limb positions, but at the expense of a high user training routine burden. The novel Composite Model concept, as presented in this study, proposed a “semi-custom fit” control solution whose concept is built upon the best aspects of both the Generalized and Personalized Model solutions combined—offering promising movement predictive accuracy, but with a reduction in required user training burden.

With this understanding, we present a conceptual illustration in Figure 4-6 of how, based on the insights from the empirical results of this work, Generalized, Personalized, and Composite Models’ movement predictive accuracies might expectedly vary over longer durations of use well beyond the time-delayed conditions investigated in this study. The accuracy values presented are

purely speculative and as such, were not derived from calculations. Figure 4-6 also presents the conceptual cumulative training routine duration over an extended period of device usage time. Figure 4-6A shows that the Generalized Model would continue to predict desired movements with a consistently low accuracy over time, under conditions of no model retraining. Alternatively, Figure 4-6B considers that if the Personalized Model was retrained when control degrades, its predicted accuracy would remain high, but at the expense of a rapidly increasing cumulative training routine burden for the user. Figure 4-6C shows that if the Personalized Model was only trained once, it would initially exhibit high predictive accuracy, but then control would degrade over time. As a compromise control solution, Figure 4-6D illustrates that the Composite Model would initially offer better movement predictive accuracy than the Generalized Model, but then would very gradually deteriorate over time in response to residual limb changes. At some point, however, Composite Model retraining would be required to reinstate the model’s initial predictive accuracy (as indicated in the upward shift in predictive accuracy in Figure 4-6D). The Composite Model’s training burden, therefore, would be somewhat aligned with that of the once-trained Personalized Model, over a considerable duration of device use. Overall, the Composite Model control concept would expectedly yield promising and sustained movement predictive accuracy,

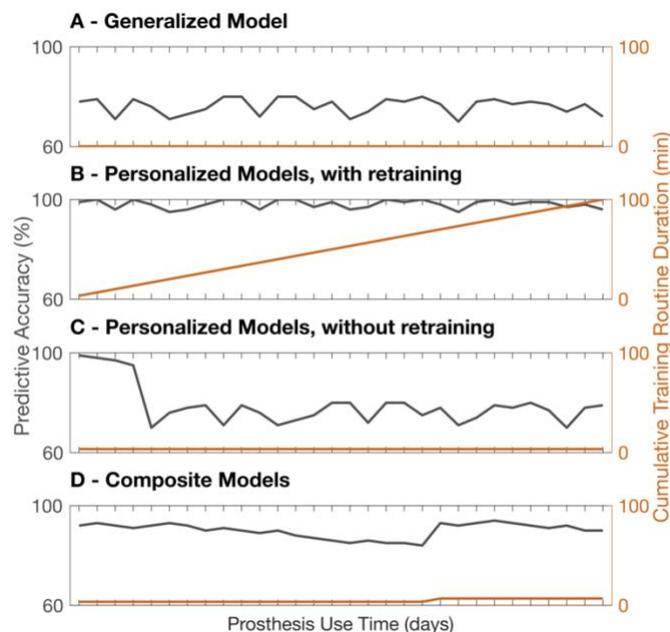


Figure 4-6. Conceptual illustration of how movement predictive accuracy might expectedly vary over extended durations of use. Conceptual predictive accuracy (grey) and cumulative training routine durations (orange) are presented for the (A) Generalized Model, (B) Personalized Models with retraining, (C) initially-trained Personalized Models without retraining, and (D) Composite Models.

with the requirement for less frequent model retraining by a user, in comparison to a Personalized Model alternative.

Ultimately, more work needs to be done to explore the movement predictive performance of the Composite Model concept for prosthesis control under conditions that require large data volumes and/or frequent model retraining by device users. Research regarding the use of a larger General Dataset to train a Generalized Model for the initial layers of a Composite Model should be undertaken. A larger General Dataset would provide a broader sampling of EMG signals and movement classes across individuals, to form the *representation* layers of a Composite Model. Furthermore, a Personalized Model could also be trained with more data. This richer dataset could be used to form a Composite Model's last *decision* layers. Such training data could be collected over multiple days, under conditions of varied levels of muscle fatigue, specific shifts in electrode positioning, and/or more limb positions. In addition, the amount of training data required by a Composite Model should be investigated in future real-time work, to quantify the optimal trade-off between model training time and desired movement predictive accuracy. An exciting future goal would see the application of the Composite Model solution used towards transfer learning-based prosthesis control. With such control, a Composite Model could be used initially, and its training data augmented with smaller subsets of data as required. Transfer learning would both lighten the user training routine load and improve predictive accuracy of prosthesis control.

This study had limitations. Notably, (1) it used a pilot study approach with a small sample size that could not yield statistically significant results, (2) the data testing was not exhaustive as only three sets of data trials were examined, and (3) only non-disabled participants were included in this work. Despite these limitations, the Composite Model concept is worthy of more rigorous examination in the future towards the reduction of user training routine burden required of pattern and deep learning-based control solutions.

## 4.5. Conclusion

This work introduced the innovative concept of a Composite RCNN-based myoelectric prosthesis control model as a means of reducing a prosthesis user's requirement for repeated execution of a long training routine while also retaining accurate predictions of user-intended movements in various limb positions. This consideration is important given that position-aware control often

requires a high volume of training data. Results of this work are anticipated to improve the usability of myoelectric devices for individuals with amputation, particularly when faced with the challenges of the *limb position effect*. As such, this research advances myoelectric control towards eliminating the need for training routine execution altogether, so that one day, a myoelectric prosthesis user may be able to immediately perform activities of daily living after donning their device.

## Chapter 5.

# RCNN Classification Control with Transfer Learning

Chapter 5 presents “A Case Series in Position-Aware Myoelectric Prosthesis Control using Recurrent Convolutional Neural Network Classification with Transfer Learning,” originally published in the Proceedings of the International Consortium on Rehabilitation Robotics, in 2023 [90]. It further investigates a novel recurrent convolutional neural network classification control solution that addresses the two important outcomes of Chapter 3—that lengthy model training routines can result in user training burden and that off-line control assessments might not represent real-use control experiences. In this current chapter, a position-aware transfer learning-based classification control solution that harnesses the benefits of transfer learning is trained and tested by participants without amputation wearing a simulated prosthesis. Note that this participant population is easier to recruit versus those with transradial amputation and has been shown to be a reasonable proxy for actual prosthesis use [77]. An important outcome of this work is the realization that existing control assessment metrics are lacking, particularly if incidents of the limb position effect problem are to be solved. An illustration of how the information presented in Chapter 5 contributes to the overall work in this thesis is presented in Figure 5-1.

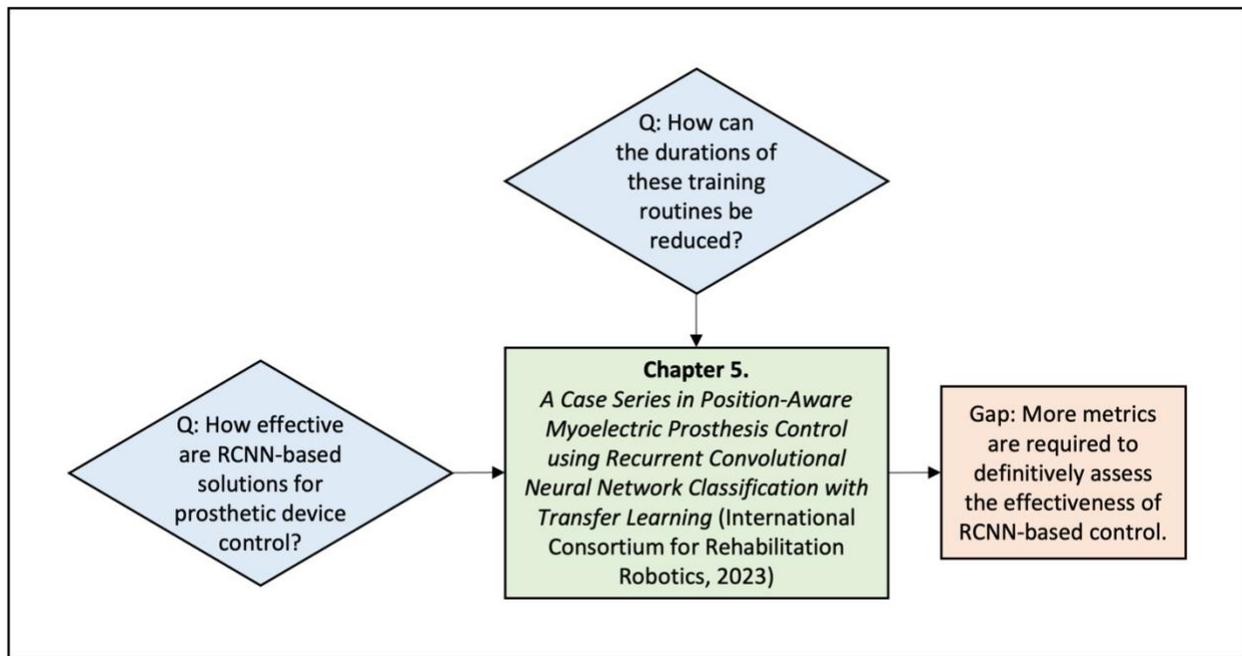


Figure 5-1. Research questions addressed by Chapter 5 and remaining gap.

## Abstract

Position-aware myoelectric prosthesis controllers require long, data-intensive training routines. Transfer Learning (TL) might reduce training burden. A TL model can be pre-trained using forearm muscle signal data from many individuals to become the starting point for a new user. A recurrent convolutional neural network (RCNN)-based classifier has already been shown to benefit from TL in *offline* analysis (95% accuracy). The present *real-time* study tested whether an RCNN-based classification controller with TL (RCNN-TL) could reduce training burden, offer improved device control (per functional task performance metrics), and mitigate what is known as the *limb position effect*. 27 participants without amputation were recruited. 19 participants performed wrist/hand movements across multiple limb positions, with resulting forearm muscle signal data used to pre-train RCNN-TL. 8 other participants donned a simulated prosthesis, retrained (calibrated) and tested RCNN-TL, plus trained and tested a conventional linear discriminant analysis classification controller (LDA-Baseline). Results confirmed that TL reduces user training burden. RCNN-TL yielded improved task performance durations over LDA-Baseline (in specific Grasp and Release phases), yet other metrics worsened. Overall, this work contributes training

condition factors necessary for TL success, identifies metrics needed for comprehensive control analysis, and contributes insights towards improved position-aware control.

## 5.1. Introduction

Individuals with a transradial amputation often use myoelectric prostheses to restore or assist their impaired upper limb function. Prosthetic devices enable users to perform everyday tasks like eating, grooming, and getting dressed. Accomplishing such tasks requires execution of prosthetic hand and wrist movements in varied limb positions. These movements are driven by motors housed within the device, with instructions sent to the motors by a controller. Wearers operate their prosthesis using residual limb muscle contractions, and surface electrodes in its socket capture resulting muscle signals using electromyography (EMG) [6]. Myoelectric controllers, including those that use pattern recognition, can interpret EMG signals. When pattern recognition is employed for control, a prosthesis wearer must perform a series of specific movements, known as a training routine, prior to using their device [7]. Once a training routine is complete, patterns evident in the captured signal features are learned by the control model, the features are classified during device use, and the resulting classifications inform motor instructions. Pattern recognition-based myoelectric controllers are commonly tested in research settings and are commercially available, but are not yet widely accepted clinically.

Research has shown that EMG signals alone might not reliably inform intended prosthesis movement, particularly during instances when a user must hold their arm in untrained positions to accomplish tasks [18]. Limb position variations can result in degraded pattern recognition-based control, as evidenced by unexpected device movements and reported user frustration [18]. This challenge is known as the *limb position effect*. To mitigate this effect, some researchers have increased the number of surface EMG electrodes worn by users (high-density electrode arrays) [58]. Other researchers have successfully introduced the addition of an inertial measurement unit (IMU), worn on a user's residual forearm for the capture of supplemental limb position data [15]. By combining EMG and IMU data, a *position-aware* pattern recognition-based controller can indeed provide reliable function across multiple limb positions [10].

To effectively mitigate the limb position effect, a control model training routine must include hand/wrist movements performed across multiple limb positions—not simply performed in a bent-

elbow position as required by conventional controllers [10]. The time and muscle activation demands of such a routine, however, become burdensome for the user [6], [11]. In addition, retraining (or calibration) is typically required of myoelectric controllers in instances when device control degrades, such as due to muscle fatigue or electrode shifts. This retraining further contributes to user training burden. Overall, mitigation of the limb position effect necessitates a burdensome and data-intensive control model solution.

Commercially available myoelectric controllers that employ pattern recognition typically use a statistical model known as linear discriminant analysis (LDA) [15], [22]. An LDA classification control model applies probability theory to discover patterns in EMG data, and then uses engineered features to inform control. Given that LDA-based control is commonly used in myoelectric prostheses, it has been adopted in research as a baseline for comparison to other controllers [10]. An emerging area of research that offers an alternative to LDA solutions uses deep learning [27]; in particular, recurrent convolutional neural networks (RCNNs) [10]. RCNNs offer the ability to learn new useful features from raw EMG signals (rather than requiring features to be extracted prior to pattern learning) and the ability to recognize the nuances of the time-varying behaviour of EMG signals [27]. RCNN classification has been investigated for position-aware prosthesis control, because it can combine large amounts of data from multiple sensors, including from EMG and IMU data streams [10].

Transfer learning (TL) is an adjunct solution that may reduce the training and retraining burden placed upon a user, as necessitated by position-aware RCNN-based classification control (but not applicable to LDA-based control). With this solution, a classification control model (classifier) can be trained using a large dataset of EMG and IMU signals obtained from numerous individuals, to become the starting point for a new user's device control. That new user would require only a reduced amount of personal movement data for training and retraining thereafter. Our earlier offline research determined that an RCNN-based classifier can indeed benefit from TL [91]. It relied on the capture of a large dataset of muscle data points for control pre-training. Our RCNN-based classifier with TL achieved 95% accuracy in movement classification. This classifier was pre-trained using data from 19 participants and required just 4 seconds of data per movement in three limb positions for retraining. The favourable results of this earlier work showed both high classification accuracy and decreased training burden.

Still, it has been shown that offline myoelectric prosthesis research outcomes do not necessarily correlate with physical device controllability [23]. Our earlier work, therefore, could be furthered through real-time research—that is, through the use of a donned simulated prosthesis. Such devices have been shown to be a good proxy for actual myoelectric prosthesis use [77]. Experimentation using a simulated device could confirm whether an RCNN-based classification controller with TL can mitigate the limb position effect and decrease training burden. To conduct this real-time research: (1) testing should include use of a donned simulated prosthesis, (2) a training routine should include multi-position hand and wrist movements, (3) functional tasks should be used for testing, and (4) established kinematic metrics should be used for outcome analysis.

The present study bridges the gap between reported offline myoelectric control outcomes and real-time device controllability when an RCNN classification controller with TL (**RCNN-TL**) is systematically compared to an LDA baseline classification controller (**LDA-Baseline**). Here, a simulated prosthesis was worn by participants without amputation, each of whom executed training routines and performed functional tasks across varying limb positions to test control. This work contributes to the literature by offering valuable lessons towards addressing the limb position effect. We investigated whether TL can indeed reduce user training burden in conjunction with RCNN-based classification. Important implications were discovered regarding the conditions required for training, and the need for comprehensive metrics to fully interpret control results.

## 5.2. Methods

### 5.2.1. Participants

Two distinct groups of participants were recruited for this study: a General Participant Group, whose data were used to pre-train RCNN-TL's model, and a Simulated Prosthesis (SP) Participant Group, who further trained and tested both RCNN-TL and LDA-Baseline. All participants provided written informed consent, as approved by the University of Alberta Health Research Ethics Board (Pro00086557).

**General Participant Group** (*without simulated prosthesis*): Nineteen participants without upper limb impairment were recruited. All had normal or corrected vision, 10 were male, nine were female, 17 were right-handed. They had a median age of 25 years (range: 19–58 years) and median height of 170 cm (range: 159–193 cm).

**SP Participant Group** (*with donned simulated prosthesis*): A total of nine new participants without upper limb impairment were recruited. One participant was removed due to their inability to reliably control the donned simulated prosthesis even after control practice. Of the remaining eight participants, all had normal or corrected vision, five were male, three were female, seven were right-handed. They had a median age of 22 years (range: 20–56 years) and median height of 181 cm (range: 169–185 cm). No participants had experience with EMG pattern recognition control using a simulated prosthesis. The eight participants completed two data collection sessions on different days (with a median of 24 days between sessions, range: 18–45 days), during which half of the participants used RCNN-TL in their first session, and the other half used LDA-Baseline in their first session. Participants used the other controller in their second session.

### 5.2.2. EMG and Accelerometer Data Collection

Each participant in both groups wore a Myo gesture control armband (Thalmic Labs, Kitchener, Canada—discontinued) at approximately the upper third of their forearm (with the top of the armband at a median of 27.83% of the way down the forearm from the medial epicondyle to ulnar styloid process), as shown in Figure 5-2A. The Myo armband contained eight surface electrodes to collect EMG data at 200 Hz. The Myo armband also contained one IMU to collect limb position data (three accelerometer, three gyroscope, and four quaternion data streams) at 50 Hz. Myo Connect software was used to stream EMG and IMU data into Matlab.

### 5.2.3. Donned Simulated Prosthesis

The simulated prosthesis used in this study was the 3D-printed Modular-Adaptable Prosthetic Platform (MAPP) [92] (shown in Figure 5-3B). It was fitted to each SP Participant Group member's right arm for simulation of transradial prosthesis use. The MAPP's previously published design [92] was altered to improve wearer comfort in our study—the distal ring was made to resemble the oval shape of a wrist and the hand brace was elongated so that the distal ring would sit more proximally on the wearer's wrist. A 3D-printed robotic hand [93] was affixed to the MAPP beneath the participant's hand. Wrist rotation capabilities were also added to the device. Hand and wrist movements were each powered by a Dynamixel MX Series motor (Robotis Inc., Seoul, South Korea).

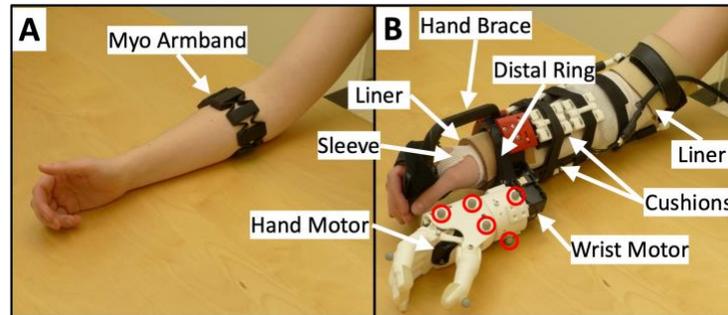


Figure 5-2. (A) Myo armband on a participant's forearm and (B) donned simulated prosthesis on a participant's forearm, with labels indicating the sleeve, 2 pieces of liner, hand brace, distal ring, cushions, wrist motor, and hand motor. Five motion capture markers are indicated with red circles.

After placement of the Myo armband, each SP Participant Group member donned a thin sleeve and then the MAPP. To increase participant comfort, pieces of thermoplastic elastomer liner were placed inside the distal ring and just above the participant's elbow, and 3D-printed cushions, made of Ninjaflex Cheetah filament (Ninjatek, Inc.), were placed throughout the device socket (shown in Figure 5-2B). The secureness of the device and each participant's comfort were checked before proceeding with controller training.

#### 5.2.4. Control Model Implementation and Training

**RCNN-TL Implementation:** Bayesian optimization automatically determined the number of convolution layers, number of filters, filter size, pooling size, and patience required for the classifier used in this controller. Optimization was performed in two steps: first, the number of layers along with each hyperparameter being optimized were determined using a broad range of values; thereafter, values were refined using a narrower range (centered at earlier optimized values). RCNN-TL's model architecture consisted of 19 layers, as illustrated in Figure 5-3. In this model, a sequence input layer first received and normalized the training data. Then, a sequence folding layer was used, allowing convolution operations to be performed independently on each window. This was followed by a block of four layers: a 2D convolution, a batch normalization, a rectified linear unit (ReLU), and an average pooling layer. This block of layers was repeated once more. Each of the two average pooling layers had a pooling size of 1x4. A block of three layers followed: a 2D convolution, a batch normalization, and a ReLU layer. The optimal number of filters in the convolution layers were determined to be 4, 16, and 32, respectively, and each had a

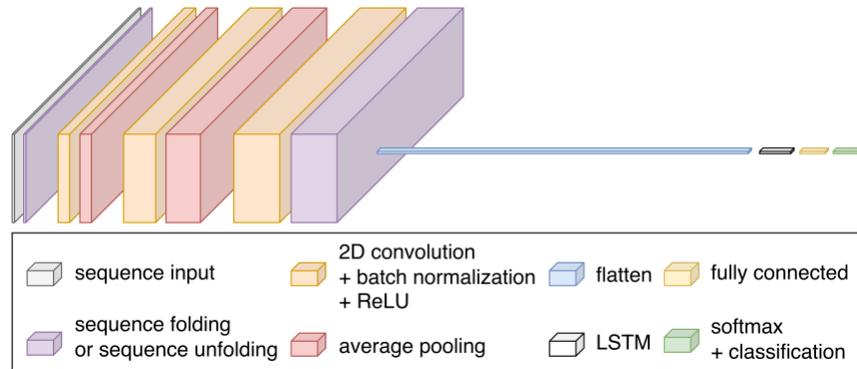


Figure 5-3. Architecture of RCNN-TL’s model: sequence input layer; sequence folding layer; two blocks of 2D convolution, batch normalization, rectified linear unit (ReLU), and average pooling; one block of 2D convolution, batch normalization, and ReLU; sequence unfolding layer; flatten layer; long short-term memory (LSTM) layer; fully connected layer; softmax layer; and classification layer.

filter window size of 1x3. The next layers included a sequence unfolding layer (to restore the sequence structure), a flatten layer, a long short-term memory (LSTM) layer, and a fully connected layer. Finally, a softmax layer and classification layer were used. To prevent overfitting, a patience parameter was set to trigger early stopping when the validation loss increased five times (e.g., similar to Côté-Allard et al. [71]).

**RCNN-TL’s Model Pre-Training Routine:** General Participant Group members followed onscreen instructions, performing muscle contractions in 5 wrist positions, for 5 seconds each: flexion, extension, pronation, supination, and rest. The muscle contractions were performed twice in 4 limb positions: arm at side, elbow bent at 90°, arm straight out in front at 90°, and arm up 45° from vertical. This position-aware routine was similar to those used in other real-time control studies aiming to mitigate the limb position effect [10], [15], [23]). The resulting EMG and accelerometer data, plus corresponding classes of muscle contractions, were used to pre-train RCNN-TL’s model.

**RCNN-TL’s Model Retraining Routine:** SP Participant Group members followed onscreen instructions, performing muscle contractions in the same 5 wrist positions, for *only 2* (rather than 5) seconds each. The muscle contractions were performed twice in *only 3* (not 4) limb positions: arm at side, elbow bent at 90°, and arm up 45° from vertical. Note that this shortened/optimized

routine was uncovered in our previous offline research [91]. The resulting EMG and accelerometer data, plus corresponding classes of muscle contractions, were used to retrain RCNN-TL's model.

**LDA-Baseline Implementation:** Four commonly used EMG features were chosen for implementation of LDA-Baseline's model: mean absolute value, waveform length, Willison amplitude, and zero crossings [9]. A pseudo-linear LDA discriminant type was used, given that columns of zeros were occasionally present in some classes for some features (including Willison amplitude and zero crossings).

**LDA-Baseline's Model Training Routine:** SP Participant Group members followed onscreen instructions, performing muscle contractions in the same 5 wrist positions, for 5 seconds each. The muscle contractions were performed twice, with the participants' elbow bent at 90°. This single-position routine mimicked standard myoelectric prosthesis training [18]. The resulting EMG data and corresponding classes of muscle contractions were used to train LDA-Baseline's model.

#### 5.2.5. Data Processing

For both model training and real-time control, the EMG data from the Myo armband were filtered using a high pass filter at 20 Hz (to remove movement artifacts), as well as a notch filter at 60 Hz (to remove electrical noise). Next, the accelerometer data streams were upsampled to 200 Hz (using previous neighbour interpolation) to align them with the corresponding EMG data. Data were then segmented into windows (160-millisecond with a 40-millisecond offset). These windows of EMG and accelerometer data were used for RCNN-TL. For LDA-Baseline, time-domain features were calculated for each EMG channel, in each window. Each model was trained in Matlab using an Intel Core i9-10900K CPU (3.70 GHz) with 128 GB of RAM. RCNN-TL's and LDA-Baseline's models were retrained/trained in median times of 3.41 and 0.39 seconds, respectively. For real-time control, the classifiers predicted wrist and hand movements in Matlab, predictions were relayed to brachI/Oplexus [94], and control signals were sent to the simulated prosthesis' motors.

### 5.2.6. Control Practice

Each SP Participant Group member took part in a control practice period. They were taught how to operate the simulated prosthesis using their muscle contractions. This control practice took approximately 40 minutes.

To determine whether participants could reliably control the simulated prosthesis, they completed an activity. Two cups were situated in front of them at two different heights, with a ball in one of the cups. Participants were asked to pour the ball between the two cups, and instances when the participants dropped the ball or a cup were recorded. If participants could not complete at least 10 pours with a success rate of at least 75% within 10 minutes, they were removed from the study. Recall that one participant was removed (as stated in Section II.A), given that they could not complete this activity with LDA-Baseline in their first session.

### 5.2.7. Motion Capture Setup

An 8-camera OptiTrack Flex 13 motion capture system (Natural Point, OR, USA) was used to capture hand movements and task objects at 120 Hz. Six individual markers were placed on the simulated prosthesis hand to ensure reliable rigid body tracking (with at least 3 markers always trackable), five of which are shown in Figure 5-2B. Note that unlabelled markers in Figure 5-2B were not used for analysis.

### 5.2.8. Functional Tasks for Control Testing

**Pasta Box Task (Pasta):** Participants were required to pick up a box of pasta and move it between a side table and two shelves at varying heights on a cart (including across their midline), and then back to the side table [95], as shown in Figure 5-4A. Motion capture markers were placed on the cart, side table, and pasta box, as per Valevicius et al. [95]. Participants performed a total of 10 Pasta trials. If participants dropped the pasta box, placed it incorrectly, performed an incorrect movement sequence, or hit the frame of the task cart, the trial was labelled as an error and not analyzed.

**Refined Clothespin Relocation Test (RCRT):** Participants were required to move three clothespins between targets on horizontal and vertical bars [96], as shown in Figure 5-4B,C. To

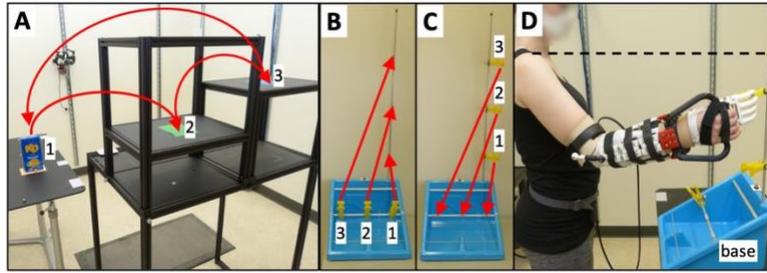


Figure 5-4. (A) Pasta Box Task trial movement order; and Refined Clothespin Relocation Test (B) Up trial movement order, and (C) Down trial movement order, and (D) adjustable cart setup.

simplify trial execution, RCRT was split into **RCRT Up** and **RCRT Down** trials. During Up trials, participants moved the clothespins from the horizontal bar (right to left positions) to the vertical bar (bottom to top positions), as shown in Figure 5-4B. During Down trials, participants moved the clothespins from the vertical bar to the horizontal bar, in the same order as in Up trials, as shown in Figure 5-4C. A height adjustable cart was set such that the top of each participants' shoulder was aligned with the midpoint between the top two targets on the vertical bar, as shown in Figure 5-4D. Five motion capture markers were placed on the cart, three on the task base, and one on each clothespin. Participants performed a total of 10 Up trials and 10 Down trials. If participants dropped a clothespin, placed it incorrectly, or performed an incorrect movement sequence, the trial was labelled as an error and not analyzed.

### 5.2.9. Experimental Data Analysis

Motion capture data analysis in this study was conducted in accordance with Valevicius et al. [95]: the marker trajectory data were cleaned and filtered; for each task, the data from each trial were divided into distinct *movements* based on hand velocity and the velocity of the pasta box/clothespins; the data from each movement were further segmented into the *phases* of Reach, Grasp, Transport, Release, and Home (the Home phase was not used for data analysis); and *movement segments* of Reach-Grasp and Transport-Release were used in hand movement analysis.

Commonly used task performance metrics were calculated as per Valevicius et al. [95]: task success rate (the percentage of trials that were error-free) was calculated for each *task*; trial duration was calculated for each *trial*; phase duration and relative phase duration were calculated

for each *phase*; and peak hand velocity, hand distance travelled, and hand trajectory variability were calculated for each *movement segment*.

To investigate task performance differences between RCNN-TL and LDA-Baseline, the following statistical analyses were performed:

**Task success rate**—Pairwise comparisons (t-test or Wilcoxon sign rank test) were conducted and deemed significant when the p value was less than 0.05.

**Trial duration**—Participants' results were averaged across trials, after which pairwise comparisons between the controllers were conducted.

**All other metrics**—Participants' results were averaged across trials and movements. If results were normally-distributed, a two-factor repeated-measures analysis of variance (RMANOVA) was conducted using the factors of controller and phase/movement segment. When the resulting controller effects or controller-phase/movement segment interactions were deemed significant (that is, when the Greenhouse-Geisser corrected p value was less than 0.05), pairwise comparisons between the controllers were conducted. If results were not normally-distributed, the Friedman test was conducted. When the resulting p value was less than 0.05, pairwise comparisons between the controllers were conducted.

### 5.3. Results

The functional task performance metrics for RCNN-TL versus LDA-Baseline are shown in Figure 5-5. Significant differences are indicated with above-bar asterisks. Improvements in task performance are characterized by high success rates, low trial durations, low phase durations, high Reach and Transport relative phase durations, low Grasp and Release relative phase durations, high peak hand velocities, low hand distances travelled, and low hand trajectory variability, as per Valevicius et al. [95].

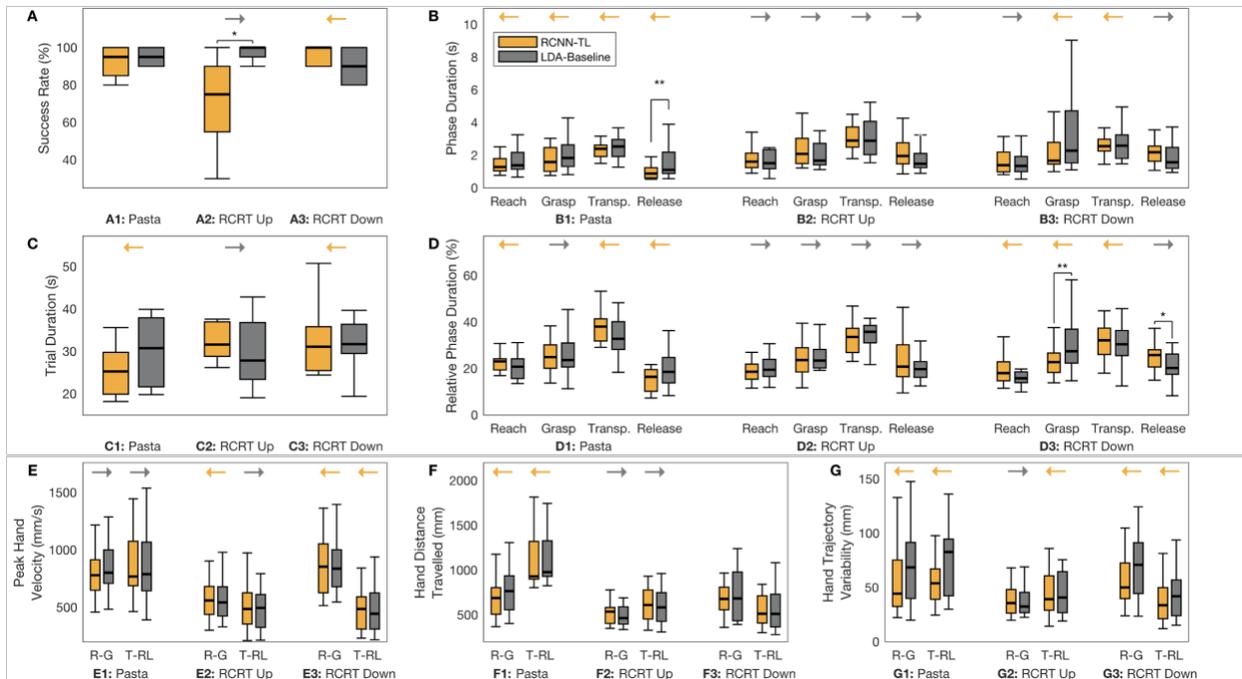


Figure 5-5. Task performance metrics results with RCNN-TL (orange) and LDA-Baseline (grey) during Pasta, RCRT Up, and RCRT Down tasks are shown for: (A) success rate, (B) phase duration, (C) trial duration, (D) relative phase duration, (E) peak hand velocity, (F) hand distance travelled, and (G) hand trajectory variability. Panels B and D present results in phases (Reach, Grasp, Transport [Transp.], Release) and Panels E–G present results in movement segments (Reach-Grasp [R-G], Transport-Release [T-RL]). Medians are indicated with thick lines, interquartile ranges are indicated with boxes, and significant differences between RCNN-TL and LDA-Baseline are indicated with asterisks (\*:  $p < 0.05$ , \*\*:  $p < 0.01$ ). Arrows indicate which controller performed better for each metric, with orange left arrows indicating RCNN-TL and grey right arrows indicating LDA-Baseline.

### 5.3.1. Significant Differences Between Controllers

Significant differences between RCNN-TL and LDA-Baseline were evident in 4 out of 48 total metrics. **RCNN-TL outperformed LDA-Baseline in 2 metrics:** Pasta Release phase duration (Figure 5-5B1) and RCRT Down Grasp relative phase duration (Figure 5-5D3). **LDA-Baseline outperformed RCNN-TL in 2 metrics:** RCRT Up success rate (Figure 5-5A2) and RCRT Down Release relative phase duration (Figure 5-5D3).

### 5.3.2. Task-Specific Observations Between Controllers

When considering *all* 48 task performance metrics (not simply those that exhibited significant differences) trends were evident for each of the three functional tasks (with 16 metrics per task).

**Pasta:** 12 of 16 metrics showed that RCNN-TL performed better (indicated by left arrows in Figure 5-5B1–D1, F1, G1); 3 of 16 metrics showed that LDA-Baseline performed better (right arrows in Figure 5-5D1, E1); and success rate showed no change (Figure 5-5A1). **RCRT Up:** 2 of 16 metrics showed that RCNN-TL performed better (left arrows in Figure 5-5E2, G2); 14 of 16 metrics showed that LDA-Baseline performed better (right arrows in Figure 5-5A2–G2). **RCRT Down:** 11 of 16 metrics showed that RCNN-TL performed better (left arrows in Figure 5-5A3–E3, G3); 3 of 16 metrics showed that LDA-Baseline performed better (right arrows in Figure 5-5B3, D3); and hand distances travelled showed only small changes (both less than 6 mm differences, Figure 5-5F3).

## 5.4. Discussion

This real-time study confirmed that TL reduces training burden when used with an RCNN-based classification controller [91]. Statistically, the functional task performance between RCNN-TL and LDA-Baseline was similar. However, compelling non-significant performance trends were identified, and many lessons were learned to direct future prosthesis control studies.

### 5.4.1. TL Impact on Training Burden

The General Participant Group members executed RCNN-TL’s full pre-training routine, which took 3.33 minutes. The SP Participant Group members, however, simply executed RCNN-TL’s 1-minute retraining routine prior to device control testing—a 70% decrease in duration. This research demonstrated that TL *is* a valuable adjunct to RCNN-based classification control, as it offers a model starting point that needs only to be calibrated using a smaller amount of individual-specific data. Notably, a TL solution is not possible with LDA-based control. To further investigate the influence of TL, we will examine an RCNN-based controller *without TL* in future work (using this study’s model architecture).

### 5.4.2. Pre-Training Conditions

Despite our promising TL-based training burden reduction results, most task performance metrics did not yield significant control improvements. A realization from this outcome points to the principle that pre-training data should be captured under conditions that closely resemble those during use. The General Participant Group members performed wrist/hand movements while not

wearing a simulated prosthesis for RCNN-TL pre-training. The SP Participant Group members, however, wore a simulated prosthesis when retraining RCNN-TL and training LDA-Baseline. The training conditions between the participant groups were somewhat dissimilar, as the donned prosthesis introduced weight, and co-activation of muscles resulted [97]. Consequently, patterns learned from the muscle signals of participants without the donned simulated prosthesis may not have optimally transferred to conditions for device use. Future research should investigate whether prosthesis weight does, in fact, play a significant role in TL-based control. A fundamental lesson learned in this study is that pre-training data should ideally be collected under physical conditions that will create the same muscle co-activation patterns exhibited during device use.

### 5.4.3. Comprehensive Metrics Needed

In our earlier offline research, RCNN-TL's model achieved a classification accuracy of 95% [91] and LDA-Baseline's model achieved a classification accuracy of 85% [10], when tested in all limb positions. In this current real-time research, only two of the 48 task performance metrics showed RCNN-TL performing significantly better than LDA-Baseline—in Pasta Release phase duration and in RCRT Down Grasp relative phase duration. At first glance, these results might seem underwhelming and may simply point towards the notion that offline results are not always indicative of real-time control performance [23]. However, interesting limb position-related findings can be surmised from this work, all pointing to a need for more comprehensive control metrics for their verification:

- (1) *RCNN-TL might offer improved position-aware control:* RCNN-TL tended to perform better than LDA-Baseline in tasks that required high limb position Grasps—instances where control expectedly deteriorates due to the limb position effect.
- (2) *A large phase duration interquartile range (IQR) might indicate a limb position effect occurrence:* An instance where the limb position effect probably occurred was during RCRT Down Grasp phases, as evidenced by a large phase duration IQR under LDA-Baseline control (3.19 s, as shown in Figure 5-5B3). This large IQR was likely due to control difficulties introduced when clothespins were grasped from the vertical bar over increasing heights. Moreover, as the same IQR was considerably smaller under RCNN-TL

control (1.33 s, shown in Figure 5-5B3), the limb position effect was seemingly mitigated by TL.

(3) *More conclusive control metrics are needed:* The metrics analyzed in this work could not definitively confirm the limb position effect instances suspected in (1) and (2) above. An examination of control characteristic metrics, such as number of grip aperture adjustments [93] and grip aperture plateau time [98], however, would offer a richer understanding of what occurred during Grasp phases and why. Furthermore, a clearer understanding of user-reported experiences would improve overall assessments of control.

## 5.5. Conclusion

The goal of position-aware myoelectric prosthesis control is to provide users with reliable device operation across *all* limb positions. As pattern recognition-based control solutions require the user to execute a training routine to inform the controller, solutions attempting to mitigate the limb position effect need lengthy and therefore burdensome routines. This research confirmed that TL can reduce such training burden. As a primary contribution, it offered important considerations for implementation of an RCNN-based classification controller with TL, be it for simulated or actual myoelectric prosthesis user research. It suggested that TL should work better when the physical conditions for pre-training and training are similar, particularly as a prosthetic device introduces muscle coactivation patterns. This work also identified the need for comprehensive metrics—to uncover control characteristics that can be mapped to user reported control experiences. Ultimately, this work offered insights towards feasible position-aware prosthesis control.

# Chapter 6.

## A Suite of Metrics for Comparative Myoelectric Prosthesis Control Research

Chapter 6 presents “A multifaceted suite of metrics for comparative myoelectric prosthesis controller research,” originally accepted for publication in PLOS One, in 2024 [99]. It addresses the need for conclusive control assessment metrics, as identified in Chapter 5. This current chapter introduces a comprehensive suite of myoelectric prosthesis control evaluation metrics that includes task performance, control characteristics, and user assessments of control. To demonstrate the suite’s value to prosthesis control research, it is deployed in a comparison of Chapter 5’s transfer learning-based classification control model versus a baseline classification model alternative. The suite proves to be a valuable research toolset, as analysis of its metrics facilitates identification of incidences and circumstances of the limb position effect problem. An illustration of how the information presented in Chapter 6 contributes to the overall work in this thesis is presented in Figure 6-1.

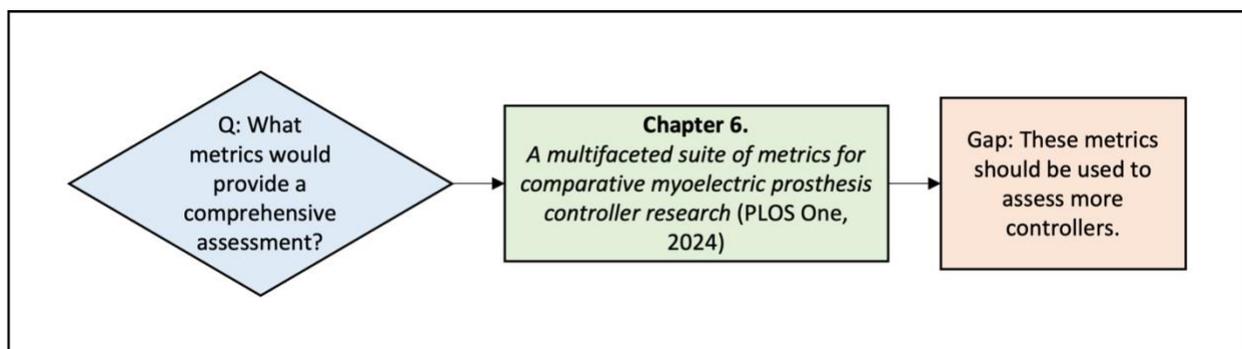


Figure 6-1. Research question addressed by Chapter 6 and remaining gap.

## Abstract

Upper limb robotic (myoelectric) prostheses are technologically advanced, but challenging to use. In response, substantial research is being done to develop person-specific prosthesis controllers that can predict a user’s intended movements. Most studies that test and compare new controllers rely on simple assessment measures such as task scores (e.g., number of objects moved across a barrier) or duration-based measures (e.g., overall task completion time). These assessment measures, however, fail to capture valuable details about: the quality of device arm movements; whether these movements match users’ intentions; the timing of specific wrist and hand control functions; and users’ opinions regarding overall device reliability and controller training requirements. In this work, we present a comprehensive and novel suite of myoelectric prosthesis control evaluation metrics that better facilitates analysis of device movement details—spanning measures of task performance, control characteristics, and user experience. As a case example of their use and research viability, we applied these metrics in real-time control experimentation. Here, eight participants without upper limb impairment compared device control offered by a deep learning-based controller (recurrent convolutional neural network-based classification with transfer learning, or RCNN-TL) to that of a commonly used controller (linear discriminant analysis, or LDA). The participants wore a simulated prosthesis and performed complex functional tasks across multiple limb positions. Analysis resulting from our suite of metrics identified 16 instances of a user-facing problem known as the *limb position effect*. We determined that RCNN-TL performed the same as or significantly better than LDA in four such problem instances. We also confirmed that transfer learning can minimize user training burden. Overall, this study contributes a multifaceted new suite of control evaluation metrics, along with a guide to their application, for use in research and testing of myoelectric controllers today, and potentially for use in broader rehabilitation technologies of the future.

### 6.1. Introduction

Below elbow (transradial) is the most prevalent of major upper limb amputations [100]. A myoelectric prosthesis offers a means of restoring complex limb function to those with transradial amputation, ideally across a wide range of arm positions [12]. Conventional myoelectric device control is based on electromyography (EMG) [101]. Here, signals are typically detected by surface

electrodes that are housed within a donned prosthesis socket and then transmitted to the device's onboard controller. The controller decodes user-specific muscle contractions and sends corresponding instructions to appropriate prosthesis wrist and hand motors.

Myoelectric prostheses that employ pattern recognition offer predictive device control that is capable of learning a user's intended movements [7], [13]. Despite the potential of such machine learning-based control solutions, device performance challenges persist for users, particularly when various limb positions are necessary [18]. In these instances, EMG signals change due to gravity, supplemental muscle activities, and electrode shifts resulting from changes in muscle topology [14]. Resulting control can be unpredictable and therefore frustrating for users [18]. This control challenge is well documented and referred to as the *limb position effect* [14]. Several pattern recognition-based control methods have been investigated to minimize the limb position effect [21], [22], [23], [60], [61], [62], [63], [64], [65], [66], [102], [103], [104], [105], [106]. These methods require a user to perform a training routine across multiple limb positions, prior to daily device use. A training routine involves execution of a specific sequence of forearm muscle contractions. EMG signals resulting from the muscle contractions are captured for use by the device controller's model. The model learns to recognize various patterns of consistent and repeated muscle signal features [7], including patterns involved in wrist rotation and hand open/close. Learned features are subsequently classified during device use, with classifications informing motor instructions.

Inertial measurement unit (IMU) data can provide a classification control model with additional and informative limb position-related data [10], [14]. Deep learning control methods, such as recurrent convolutional neural networks (RCNNs), can combine high volumes of EMG and IMU data from multiple limb positions. However, to capture all required muscle and limb position data (in low to high arm positions), lengthy and burdensome training routines must be performed by users [6], [11], [12]. Control model retraining is also required in instances when device control degrades, such as due to muscle fatigue or electrode shifts. The overall training burden poses drawbacks to position-aware myoelectric control methods.

Our earlier study uncovered that the addition of transfer learning (TL) can alleviate the training burden necessitated by data intensive RCNN-based solutions [90]. In this previous work, an RCNN classification control model (classifier) was trained using a large dataset of EMG and IMU signals

obtained from numerous individuals with intact upper limbs, to become the starting point of new users' control. Each new user required a reduced amount of personal data for training thereafter. To test control, participants wearing a simulated prosthesis [77], [107] performed two functional tasks: the Pasta Box Task [95] and the Refined Clothespin Relocation Test [96]. Control assessments were based on metrics established in the literature [77], [95], [107], [108], [109]. This research showed that RCNN-based classification control with TL reduces training burden and offers a control solution with a tendency towards better functional task performance across multiple limb positions (versus a linear discriminant analysis, or LDA, classification controller). Interestingly, the research also identified possible instances of the limb position effect during high grasping movements, however it was noted that more detailed measures of control were needed to confirm this [90]. As a corollary to the TL-based findings resulting from this work, metrics deficiencies were uncovered—control characteristics outcomes evidenced during task performance could not be fully understood, and user-reported control experiences were not considered [90].

Without question, our earlier work identified important omissions in the collective of metrics commonly used for prosthesis control appraisal, even amongst those of established task-based assessments meant to mimic activities of daily living. Such functional assessments include the Box and Blocks Test [110], Jebsen-Taylor Hand Function Test [111], Activities Measure for Upper Limb Amputees [112], Southampton Hand Assessment Procedure [113], and Assessment of Capacity for Myoelectric Control [114], [115]. These assessments typically require users to interact with a variety of objects (e.g., grasp, move, rotate, release of objects), but the scores used to summarize arm function are limited—either based on task completion durations [111], [113], number of objects moved [110], or a trained rater's assessment [112], [114], [115]. Such scores cannot yield a complete understanding of the quality of participants' hand, wrist, and arm movements [95]. Furthermore, they cannot adequately characterize the nature of device control, such as the identification of unnecessary grip aperture adjustments [90]. Even a recently introduced take-home assessment method left researchers unable to distinguish how available grips were used and how the effects of usage conditions affected control [116]. For all such reasons, we determined that RCNN-based classification control with TL (RCNN-TL) not be judged in future work by task performance alone, but rather that control characteristics also be considered. Then collectively, the

task performance and control characteristics can be weighed against subjective user experience, to provide a full complement of data-driven prosthesis control outcomes.

This current work re-examines RCNN-TL control through the lens of comprehensive assessment metrics. As a primary contribution, it introduces a novel suite of metrics that aims to address the issue of inconclusive myoelectric controller assessment outcomes. The suite includes three broad categories of metrics: **task performance**, **control characteristics**, and **user experience**. As a secondary contribution, this work showcases the use and thoroughness of the suite by deploying these metrics to reinvestigate our earlier controller research findings [90]—examining whether RCNN-TL can indeed reduce training burden, offer improved device control over a comparative LDA baseline classification controller (LDA-Baseline), and if instances of the limb position effect can be pinpointed. In using the suite of metrics, this work provides a data-driven understanding of *when* and *why* TL-based neural network control solutions show great promise towards solving the limb position effect challenge. It is expected that the suite introduced by this research will guide future rehabilitation device control experimentation.

What follows is a presentation of our collective of metrics, an overview of our reinvestigation research, and a detailed presentation of its methods, results, discussion, and conclusion.

## 6.2. Introducing the Suite of Control Evaluation Metrics

Performance-based assessments for the evaluation of real-time upper limb prosthesis control often require participants to either move a virtual arm to a target posture or a cursor to a target position, by performing appropriate muscle contractions in their residual limb [23], [40], [42], [43], [106], [117], [118], [119], [120], [121], [122], [123], [124]. For example, a virtual arm is presented on a computer screen in the Target Achievement Control test [125], and a cursor is presented on-screen in Fitts' Law tests [126]. EMG sensors placed on participants' limbs record the corresponding muscle signals in such assessments. However, limb kinematics and factors that change EMG signals (including the limb position effect) are not taken into consideration [117]. Consequently, research that employs on-screen assessments often recommend that future work be undertaken using alternative real-time methods [23], [124].

Other studies have taken a next step towards a deeper understanding of control, through the introduction of functional task assessments [35], [114], [115], [122], [127], [128], [129], [130],

[131], [132], [133], [134], [135], [136]. Here, either non-disabled participants wearing a simulated prosthesis or actual myoelectric prosthesis users, are required to perform upper limb tasks that mimic activities of daily living. One common clinical control assessment technique, known as the Assessment of Capacity for Myoelectric Control, has a trained rater assess control during functional task execution [114], [115]. However, a trained rater may not always be available. Alternative non-rater-based assessment approaches record the movement of participants' upper limbs during task execution, using motion capture technology. From the resulting data, hand movement metrics, including hand velocity, hand distance travelled, and hand trajectory variability can be calculated (rather than rated) [95], [109], [137], [138]. As we had motion capture technology available to us, we used these three hand movement metrics, plus common task success rate and duration-based measures—collectively presenting them as **Task Performance Metrics**. All such metrics can be found in Table 6-1, with calculations derived from the work of Valevicius et al. [95]. Note that these metrics have already been validated [109] and have been employed to examine both myoelectric and body-powered prosthesis use [77], [108]. Furthermore, they have been compared to common clinical assessments with prosthesis users [139], [140]. Appendix C offers select examples of strong and weak outcomes resulting from these metrics.

Task performance metrics alone, however, do not capture the nuances of device control characteristics, such as instances where a user introduces unnecessary hand/wrist movements when grasping or releasing an object [90]. Some studies have introduced metrics that quantify specific myoelectric prosthesis control characteristics, including misclassification rates [141], grasp force [141], [142], grip aperture plateau time [98], wrist rotation range of motion [141], workload (assessed via pupil size) [143], and measures of muscle activations [144]. We selected and derived metrics from the literature, plus developed additional novel metrics, and collectively present them as **Control Characteristic Metrics**. All such metrics can be found in Table 6-2, with select examples in Appendix C.

Finally, whether any proposed control solution yields noticeable improvement depends on users' assessments. The National Aeronautics and Space Administration Task Load Index (NASA-TLX) is a survey tool, which measures subjective mental workload [145]. A recent literature review confirmed that the NASA-TLX has been widely employed in prosthesis use assessments [146]. Usability surveys offer yet another assessment approach and capture the users' opinions of

alternative device control solutions [147]. We selected relevant survey questions from the literature and present them as **User Experience Metrics**. All such metrics can be found in Table 6-3.

Table 6-1, Table 6-2, and Table 6-3, collectively describe the **Suite of Myoelectric Control Evaluation Metrics** introduced in this work. The Control Improvement Indicator column in each of these tables uses unimpaired limb movement as a yardstick for control assessment. To conduct such assessments, data collection protocols should include the following:

- Control models under investigation should be trained by participants using a research-specific series of hand/wrist movements that elicit forearm muscle signals for capture
- Participants must either wear a myoelectric prosthesis or a simulated prosthesis
- Participants must perform functional task(s) that can be split into the distinct phases of Reaches, Grasps, Transports, and Releases. All such task(s) should be standardized—using specific object sizes and locations for grasp and release actions. Examples of suitable functional tasks include, but are not limited to, the Pasta Box Task [95], Cup Transfer Task [95], Refined Clothespin Relocation Test [96], and the modified Box and Blocks test [148], [149].
- Motion capture data, muscle signals, and device motor data should be collected during functional task execution
- User experience survey responses should be collected at the end of each testing session
- Data streams of interest (for each functional task trial) could include: the number of error-free trials executed; trial time stamps; the 3D position of the device hand, its grip aperture and wrist rotation angles; plus the participants' shoulder flexion/extension angles, EMG signal data, and post-testing session survey scores.
- Given that standardized tasks are to be used, results calculated with the suite of metrics should only be compared within each task, not across tasks

With adherence to the above-mentioned data collection requirements, the suite of control evaluation metrics presented in this work facilitates in-depth analysis that will uncover numerous upper limb prosthesis control insights. These insights are expected to be particularly beneficial in investigations that compare myoelectric device controllers.

Table 6-1. Description of Task Performance metrics used in analysis. For each metric, the following details are outlined: the metric’s name; a description of the metric; the data analysis level at which the metric is calculated (Controller, Task, Trial, Movement, Movement Segment, or Phase); the data required for the metric calculation; the metric calculation procedure; and indicators that constitute a control improvement.

Metric	Metric Description	Metric Calculation			Control Improvement Indicator (∴ means because)
		Data Analysis Level	Required Data	Calculation Procedure	
Success Rate	Percent of trials that are error-free (%)	Task	Number of error-free trials	The number of error-free trials divided by the number of possible trial attempts [95]	A higher success rate ∴ participants make fewer errors
Trial Duration	Elapsed time for each trial (sec)	Trial	Timestamps	The timestamp at the end of the trial's last phase minus the timestamp at the start of trial's first phase [95]	A shorter trial duration ∴ participants complete trials quicker, also indicative of higher skill level [77], [98]
Phase Duration	Elapsed time for each phase (sec)	<i>Phase:</i> Reach, Grasp, Transport, Release	Timestamps	The timestamp at the end of the phase minus the timestamp at the beginning of the phase [95]	A shorter phase duration ∴ participants complete phases quicker, also indicative of higher skill level [77], [98]
Relative Phase Duration	Elapsed time for each phase, relative to the elapsed time for a Reach-Grasp-Transport-Release movement (%)	<i>Phase:</i> Reach, Grasp, Transport, Release	Timestamps	The phase duration divided by the movement duration (i.e., the difference between the timestamps at the beginning and end of a Reach-Grasp-Transport-Release movement) [95]	For Grasp and Release phases: A shorter relative phase duration ∴ participants manipulate objects with more ease [77] For Reach and Transport phases: Shorter Grasp and Release relative phase durations lead to longer Reach and Transport relative phase durations [77]
Peak Hand Velocity	Maximum velocity of the hand while moving (mm/s)	<i>Movement Segment:</i> Reach-Grasp, Transport-Release	3D Position of the Hand (mm, mm, mm)	The maximum hand velocity achieved within a movement segment [95]	A higher peak hand velocity ∴ participants move their hand quicker
Hand Distance Travelled	Total distance travelled by the hand while moving (mm)	<i>Movement Segment:</i> Reach-Grasp, Transport-Release	3D Position of the Hand (mm, mm, mm)	Calculation steps: 1. The distance between the 3D positions of the hand at successive sampled points in time are calculated 2. The sum of these distances is calculated [95]	A shorter hand distance travelled ∴ the hand movement paths of participants are more efficient [77]
Hand Trajectory Variability	How much the hand movement path varies between trials (mm)	<i>Movement Segment:</i> Reach-Grasp, Transport-Release (Note that hand trajectory variability is calculated for each participant, rather than for each trial)	3D Position of the Hand (mm, mm, mm)	Calculation steps, used for each participant: 1. The 3D position (x, y, z) of the hand is time-normalized to 101 sampled points in time (0%, 1%, ..., 100% of the movement segment) for each of the participant's trials 2. At each sampled point, the standard deviation of the x, y, and z positions across the participant's trials are independently calculated 3. The mean of the resulting standard deviations is calculated at each sampled point 4. The maximum of these means is determined [95]	A smaller hand trajectory variability ∴ participants do not explore a variety of hand movement paths (possibly indicating confidence in their chosen path with movements in keeping with those of unimpaired individuals) [77], [95]

Table 6-2. Description of Control Characteristics metrics used in analysis. For each metric, the following details are outlined: the metric’s name; a description of the metric; the data analysis level at which the metric is calculated (Controller, Task, Trial, Movement, Movement Segment, or Phase); the data required for the metric calculation; the metric calculation procedure; and indicators that constitute a control improvement. (Table 6-2 found on next page)

Metric	Metric Description	Metric Calculation			Control Improver Indicator (∴ means because)
		Data Analysis Level	Required Data	Calculation Procedure	
Total Grip Aperture Movement	Total amount of grip aperture variation (mm)	<i>Phase:</i> Reach, Grasp, Transport, Release	Grip Aperture (mm)	1. The absolute differences between the grip apertures at successive sampled points in time are calculated 2. The sum of these differences is calculated	A smaller total grip aperture movement ∴ fewer unnecessary hand open/close movements occur
Total Wrist Rotation Movement	Total amount of wrist rotation angle variation (deg)	<i>Phase:</i> Reach, Grasp, Transport, Release	Wrist Rotation Angle (degree)	1. The absolute differences between the wrist rotation angles at successive sampled points in time are calculated 2. The sum of these differences is calculated	A smaller total wrist rotation movement ∴ fewer unnecessary wrist rotation movements occur
Number of Grip Aperture Adjustments	Number of times that grip aperture variation commences or changes direction	<i>Phase:</i> Reach, Grasp, Transport, Release (Note that Transport adjustments may not cause object drop)	Grip Aperture (mm)	1. The grip aperture velocity is smoothed using a moving average filter 2. The sign of the smoothed velocity is used to detect the following: a change from 0 to positive; a change from 0 to negative; a change from negative to positive; a change from positive to negative 3. The number of times that the above sign changes occurred is tallied [93]	A smaller number of grip aperture adjustments ∴ fewer misclassifications are made [93]
Number of Wrist Rotation Adjustments	Number of times that wrist rotation angle variation commences or changes direction	<i>Phase:</i> Reach, Grasp, Transport, Release	Wrist Rotation Angle (degree)	The number of wrist rotation adjustments were calculated as per grip aperture velocity detailed above (but with wrist rotation data rather than grip aperture data).	A smaller number of wrist rotation adjustments ∴ fewer misclassifications are made [93]
Grip Aperture Plateau	Amount of time during which the grip aperture remains open before closing to grasp a task object (sec)	<i>Movement Segment:</i> Reach-Grasp	Grip Aperture (mm)	1. The grip aperture and grip aperture velocity are both smoothed using a moving average filter 2. Plateaus are identified when both of the following conditions are met: - the smoothed grip aperture is greater than 90% of the maximum grip aperture in the movement segment - the smoothed grip aperture velocity is less than 10% of the maximum grip aperture velocity in the movement segment 3. The sum of the time duration(s) in which the above conditions are met is calculated (modified from Bouwsema et al.'s calculation [98])	A shorter plateau ∴ more natural movements are made (more closely mimicking those of unimpaired individuals) and higher skill level is evident [77], [98]
Simultaneous Wrist-Shoulder Movements	Percent of the phase during which the wrist rotation is controlled while the shoulder is moving (%)	<i>Phase:</i> Any phase that requires both wrist rotation and shoulder movements.	Wrist Rotation Angle (degree) & Shoulder Flexion/Extension Angle (degree)	1. Wrist rotation and shoulder flexion/extension angular velocities are smoothed using a moving average filter 2. The absolute values of the smoothed angular velocities are calculated 3. Simultaneous movements are identified when both of the following conditions are met: - The smoothed wrist rotation angular velocity is above a threshold of 10% of the maximum wrist rotation angular velocity in the movement - The smoothed shoulder flexion/extension angular velocity is above a threshold of 10% of the maximum shoulder flexion/extension angular velocity in the movement 4. The sum of the time duration(s) in which the above conditions are met is calculated 5. The resulting sum is divided by the phase duration (i.e., the timestamp at the end of the phase minus the timestamp at the beginning of the phase)	A larger percent of simultaneous wrist-shoulder movements ∴ participants control wrist rotation while moving their arm, rather than pausing arm movements to focus on device control—in keeping with unimpaired movements
Total Muscle Activity	Total amount of muscle activity expended	<i>Phase:</i> Reach, Grasp, Transport, Release	EMG signals (range of -1 to 1)	1. The EMG signals are filtered and rectified 2. Each EMG channel is normalized using training data: a) signals from the rest wrist position with the elbow bent at 90° are obtained from that session's training data; b) step 2a rest signals are rectified; c) the mean of the step 2b rectified signals is calculated for each channel; d) each channel's mean is subtracted from that channel's step 1 signals 3. Step 2 signals are summed across channels 4. The envelope of the step 3 signals is obtained 5. The step 4 envelope is summed across time (similar to the calculation of Ingraham et al.'s composite sum EMG metric [144])	A smaller value ∴ participants exert less muscle activity

Table 6-3. Description of User Experience metrics used in analysis. For each metric, the following details are outlined: the metric’s name; a description of the metric; the data analysis level at which the metric is calculated (Controller, Task, Trial, Movement, Movement Segment, or Phase); the data required for the metric calculation; the survey question; and indicators that constitute a control improvement.

Metric	Metric Description		Metric Calculation			Control Improvement Indicator (∴ means because)
			Data Analysis Level	Required Data	Survey Question	
NASA-TLX	Mental Demand	Workload demand resulting from each controller	Controller	Survey Scores (rating 0–100)	How much mental and perceptual activity was required (e.g., thinking, remembering, looking, etc.)? Was the control easy or demanding, simple or complex? [145]	A smaller score ∴ participants experience less workload demand
	Physical Demand				How much physical activity was required (e.g., very strong muscle contractions, upper arm movements, trunk movements, etc.)? Was the control easy or demanding, restful or laborious? [145]	
	Temporal Demand				How much time pressure did you feel to complete the tasks with this control strategy? Was the pace slow and leisurely or rapid and frantic? [145]	
	Performance				How hard did you have to work (mentally & physically) to accomplish your level of performance? [145]	
	Effort				How successful do you think you were in accomplishing the goals of the task with this control strategy? How satisfied were you with your performance in accomplishing these goals? [145]	
	Frustration				How insecure, discouraged, irritated, stressed, and/or annoyed versus secure, gratified, content, relaxed, and/or complacent did you feel during the task? [145]	
Usability Survey	Intuitiveness	Usability of each controller	Controller	Survey Scores (rating 0–5)	Intuitiveness is defined as how easy it was to learn how to use the controller.	A larger score ∴ participants report greater ease of use (device controllability)
	Task Effectiveness				How easy is the controller to learn? [147]	
	Reliability				Effectiveness is defined as how well the controller was able to perform the task.	
					How well did the controller perform the task? [147] ( <i>asked for each task performed</i> )	
	Reliability is defined as how often the controller did something that was intended or expected.					
	How often did the arm move in a way that you wanted or expected? [147]					

### 6.3. Overview: Reinvestigation using our Suite of Metrics

As an example of how the suite of evaluation metrics introduced in this work can be used to advance prosthesis control research, we deployed them in a deliberately challenging experiment—reinvestigating our earlier comparative classifier research findings [90]. Here, device control offered by two classifiers was compared: a proposed deep learning-based controller (RCNN-TL) intent on mitigating the limb position effect, versus a commonly used and commercially available controller (LDA-Baseline). Given that LDA-based control is commonly used in myoelectric prostheses, it has been adopted in research as a baseline for comparison to other controllers [10],

[25], [26]. Figure 6-2 presents an overview of how each control model was trained and tested, using two distinct groups of participants without upper limb impairment who wore an EMG and IMU data capture armband.

(1) A large General Participant Group's data created a control starting point for new users.

**RCNN-TL's Model Pre-Training**—Each General Participant Group member performed a training routine (isotonic forearm muscle contractions were executed in four limb positions), during which their forearm EMG and IMU signals were collected. Their collective, resulting signal data, along with the corresponding classes of muscle contractions, informed RCNN-TL's pre-trained control model.

(2) A new, smaller Simulated Prosthesis (SP) Participant Group wore a simulated prosthesis.

**RCNN-TL's Model Retraining & Testing**—Each SP Participant Group member performed a brief training routine (isometric contractions were held in three limb positions). The resulting participant-specific EMG and IMU data, plus classes of muscle contractions, were used to calibrate RCNN-TL's model. Participants tested RCNN-TL by performing functional tasks across multiple limb positions—the Pasta Box Task [95] and the Refined Clothespin Relocation Test (RCRT) [96].

**LDA-Baseline's Model Training & Testing**—The forearm muscle signals of each SP Participant Group member were also captured using a standard pattern recognition training routine that was not designed to mitigate the limb position effect (isometric contractions were held in one limb position) [10]. The resulting EMG data, plus classes of muscle contractions, were used to train LDA-Baseline's model. Each SP Participant Group member tested LDA-Baseline by performing the Pasta Box Task [95] and RCRT [96].

## 6.4. Methods

What follows are details about our reinvestigation research methods, including: participant descriptors; muscle signal data collection and processing techniques; a description of the simulated prosthesis donned by participants; specifications of the control models under investigation and their training requirements; setup of the testing environment; the functional tasks used to assess control; the participant survey administration process; control data processing techniques to yield

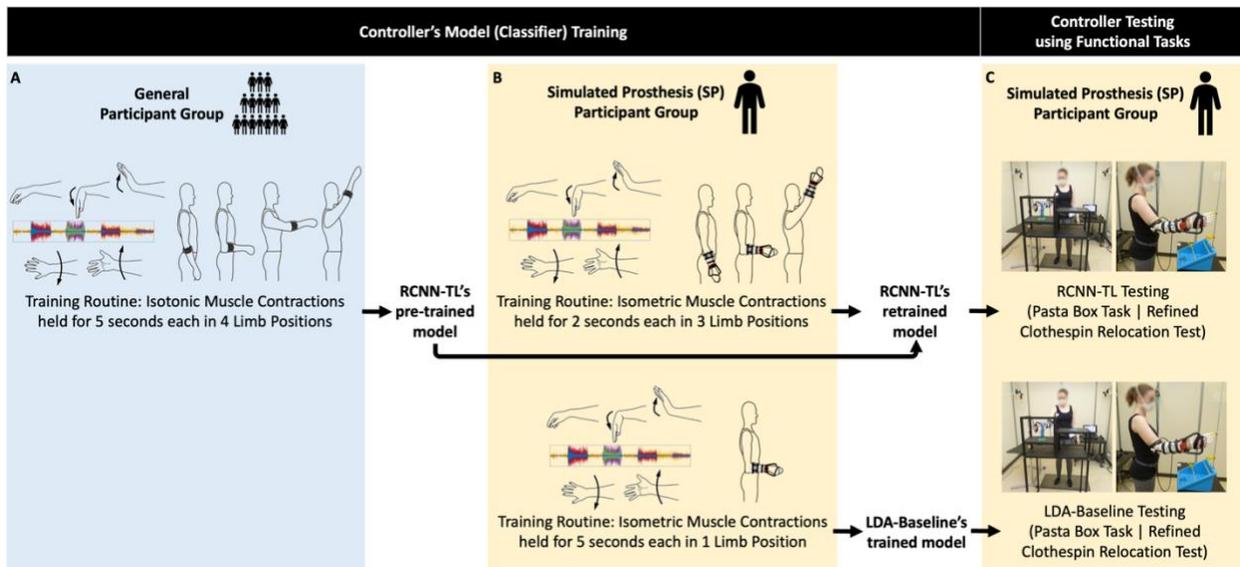


Figure 6-2. RCNN-TL and LDA-Baseline's model training and testing. The blue panel (A) illustrates the step that the General Participant Group performed (training routine that yielded RCNN-TL's pre-trained model) while wearing an EMG and IMU armband. The yellow panels illustrate the steps that the Simulated Prosthesis (SP) Participant Group performed: (B) respective training routines that yielded RCNN-TL's retrained model and LDA-Baseline's trained model, and (C) subsequent controller testing using functional tasks, all while wearing an EMG and IMU armband plus a simulated prosthesis.

the suite of metrics; the statistical analysis of such metrics; and the identification of instances of the limb position effect resulting from this analysis.

#### 6.4.1. Participants

Participants recruitment took place from March 2, 2022, to March 31, 2022. All participants provided written informed consent, as approved by the University of Alberta Health Research Ethics Board (Pro00086557).

**General Participant Group** (*without simulated prosthesis*)—Nineteen participants without upper limb impairment were recruited. All had normal or corrected vision, 10 were male, nine were female, 17 were right-handed. They had a median age of 25 years (range: 19–58 years) and median height of 170 cm (range: 159–193 cm). Each of the 19 participants completed one data collection session.

**SP Participant Group** (*with donned simulated prosthesis*)—A total of nine new participants without upper limb impairment were recruited. One participant was removed due to their inability

to reliably control the donned simulated prosthesis even after control practice. Of the remaining eight participants, all had normal or corrected vision, five were male, three were female, seven were right-handed. They had a median age of 22 years (range: 20–56 years) and median height of 181 cm (range: 169–185 cm). No participants had experience with EMG pattern recognition control using a simulated prosthesis. The eight participants completed two data collection sessions on different days, with a median of 24 days between sessions (range: 18–45 days). Half of the participants retrained/tested RCNN-TL in their first (as shown in Figure 6-2B–C), and the other half trained/tested LDA-Baseline in their first session (also shown in Figure 6-2B–C). Each trained/tested the other controller in their second session.

#### 6.4.2. Signal Data Collection & Processing Procedure

Participants in both groups wore a Myo gesture control armband (Thalmic Labs, Kitchener, Canada) over their largest forearm muscle bulk [15]. That is, at approximately the upper third of their forearm, as shown in Figure 6-3A (with the top of the armband at a median of 28% of the way down the forearm from the medial epicondyle to the ulnar styloid process). The Myo armband contained eight surface electrodes to collect EMG data at a sampling rate of 200 Hz. The Myo armband also contained one IMU to collect limb position data (three accelerometer, three gyroscope, and four quaternion data streams) at 50 Hz. Myo Connect software was used to stream and record EMG and IMU data in Matlab.

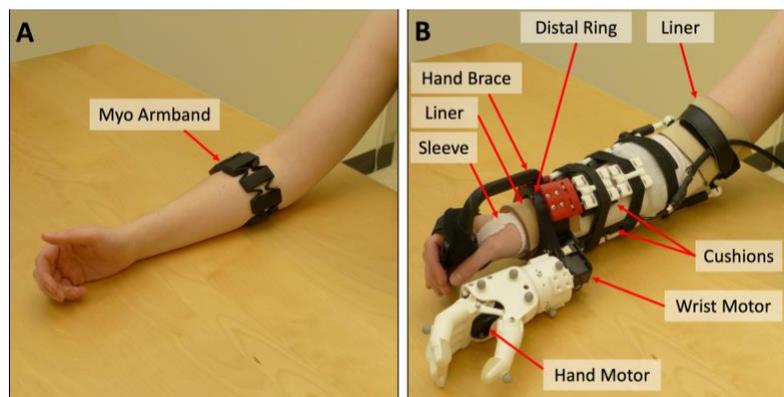


Figure 6-3. A) Myo armband on a participant's forearm and B) simulated prosthesis on a participant's forearm, with labels indicating the sleeve, two pieces of liner, hand brace, distal ring, cushions, wrist motor, and hand motor. Adapted from Williams et al. [90].

The EMG data from the Myo armband were filtered using a high pass filter with a cutoff frequency of 20 Hz (to remove movement artifacts), as well as a notch filter at 60 Hz (to remove electrical noise). The accelerometer data streams were upsampled to 200 Hz (using previous neighbour interpolation) to align them with the corresponding EMG data. Data were then segmented into windows (160-millisecond with a 40-millisecond offset).

### 6.4.3. Simulated Prosthesis & Donning Procedure

The simulated prosthesis used in this study was the 3D-printed Modular-Adaptable Prosthetic Platform (MAPP) [92] (shown in Figure 6-3B). It was fitted to each SP Participant Group member's right arm for simulation of transradial prosthesis use. The MAPP's previously published design [92] was altered to improve wearer comfort in our study—the distal ring was made to resemble the oval shape of a wrist and the hand brace was elongated so that the distal ring would sit more proximally on the wearer's wrist. A nonproprietary 3D-printed robotic hand [93] was affixed to the MAPP beneath the participant's hand. Wrist rotation capabilities were also added to the device. Hand and wrist movements (that is, with two degrees of freedom) were powered by two Dynamixel MX Series motors (Robotis Inc., Seoul, South Korea).

After placement of the Myo armband, each SP Participant Group member donned a thin, protective sleeve and then the simulated prosthesis. To increase participant comfort, pieces of thermoplastic elastomer liner were placed inside the distal ring and just above the participant's elbow, and 3D-printed cushions, made of Ninjaflex Cheetah filament (Ninjatek, Inc.), were placed throughout the device socket (shown in Figure 6-3B). The secureness of the device and the participants' comfort were checked before proceeding with controller training.

### 6.4.4. Control Model Descriptions & Training Routines

**RCNN-TL's Model**—Bayesian optimization automatically determined the number of convolution layers, number of filters, filter size, pooling size, and patience required for the classifier used in this controller. Optimization was performed in two steps: first, the number of layers along with each hyperparameter being optimized were determined using a broad range of values; thereafter, values were refined using a narrower range (centered at earlier optimized values). RCNN-TL's model architecture consisted of 19 layers, as illustrated in Figure 6-4. In this model, a sequence input layer first received and normalized the training data. Then, a sequence

folding layer was used, allowing convolution operations to be performed independently on each window of EMG and accelerometer data. This was followed by a block of four layers: a 2D convolution, a batch normalization, a rectified linear unit (ReLU), and an average pooling layer. This block of layers was repeated once more. Each of the two average pooling layers had a pooling size of 1x4. A block of three layers followed: a 2D convolution, a batch normalization, and a ReLU layer. The optimal number of filters in the convolution layers were determined to be 4, 16, and 32, respectively, and each had a filter window size of 1x3. The next layers included a sequence unfolding layer (to restore the sequence structure), a flatten layer, a long short-term memory (LSTM) layer, and a fully connected layer. Finally, a softmax layer and classification layer were used. To prevent overfitting, a patience parameter was set to trigger early stopping when the validation loss increased five times (similar to methods used in other works, including Côté-Allard et al. [71]).

**RCNN-TL’s Model Pre-Training Routine**—General Participant Group members followed onscreen instructions, performing muscle contractions in 5 wrist positions (rest, flexion, extension, pronation, and supination; shown in Figure 6-2A), for 5 seconds each. The muscle contractions were performed twice in 4 limb positions: arm at side, elbow bent at 90°, arm out in front at 90°, and arm up 45° from vertical (shown in Figure 6-2A). This position-aware routine was similar to those used in other real-time control studies aiming to mitigate the limb position effect [10], [15], [23]) and took approximately 200 seconds. The resulting EMG and accelerometer data, plus corresponding classes of muscle contractions, were used to pre-train RCNN-TL’s model.

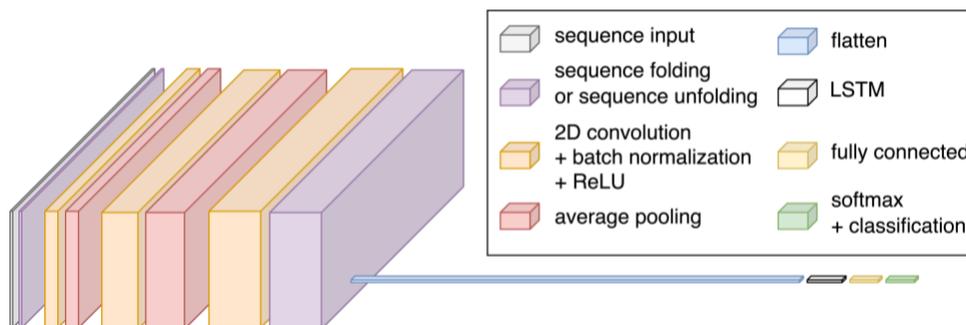


Figure 6-4. Architecture of RCNN-TL’s model: sequence input layer; sequence folding layer; two blocks of 2D convolution, batch normalization, rectified linear unit (ReLU), and average pooling; one block of 2D convolution, batch normalization, and ReLU; sequence unfolding layer; flatten layer; long short-term memory (LSTM) layer; fully connected layer; softmax layer; and classification layer. Adapted from Williams et al. [90].

**RCNN-TL’s Model Retraining Routine**—Our previous offline research [91] examined methods of reducing user training burden and uncovered a shortened/optimized routine that still yielded high predictive accuracy. In keeping with this, the SP Participant Group members followed onscreen instructions, performing muscle contractions in the same 5 wrist positions (shown in Figure 6-2B), for *only 2* (rather than 5) seconds each. The muscle contractions were performed twice in *only 3* (not 4) limb positions: arm at side, elbow bent at 90°, and arm up 45° from vertical (shown in Figure 6-2B). The resulting EMG and accelerometer data, plus corresponding classes of muscle contractions, were used to retrain RCNN-TL’s model.

**LDA-Baseline’s Model**—Four commonly used EMG features were chosen for implementation of this controller’s classifier: mean absolute value, waveform length, Willison amplitude, and zero crossings [9]. These features were calculated for each channel within each window of EMG data. A pseudo-linear LDA discriminant type was used, given that columns of zeros were occasionally present in some classes for some features (including Willison amplitude and zero crossings).

**LDA-Baseline’s Model Training Routine**—SP Participant Group members followed onscreen instructions, performing muscle contractions in the same 5 wrist positions (shown in Figure 6-2B), for 5 seconds each. The muscle contractions were performed twice, with the participants’ elbow bent at 90° (shown in Figure 6-2B). This single-position routine mimicked standard myoelectric prosthesis training [18] and took approximately 50 seconds. The resulting EMG data and corresponding classes of muscle contractions were used to train LDA-Baseline’s model.

**RCNN-TL & LDA-Baseline Implementation**—Each model was trained using Matlab software running on a computer with an Intel Core i9-10900K CPU (3.70 GHz) with 128 GB of RAM. RCNN-TL’s and LDA-Baseline’s models were retrained/trained in median times of 3.41 and 0.39 seconds, respectively. For both controllers, Matlab code was written to receive signal data and subsequently classify wrist movements. Code was also written to send motor instructions, based on the resulting classifications, to brachI/Oplexus software [94] (flexion controls hand close, extension controls hand open, pronation controls wrist counter-clockwise rotation, and supination controls wrist clockwise rotation). brachI/Oplexus relayed the corresponding control signals to the simulated prosthesis’ motors. The positions of the motors were recorded with a sampling rate of 50 Hz.

#### 6.4.5. Simulated Device Control Practice & Testing Eligibility

During each testing session, SP Participant Group members took part in a control practice period (approximately 40 minutes), during which they were taught how to operate the simulated prosthesis using isometric muscle contractions, under three conditions:

- (1) Controlling the hand open/close while the wrist rotation function was disabled. They practiced grasping, transporting, and releasing objects at varying heights.
- (2) Controlling wrist rotation while the hand open/close function was disabled. They practiced rotating objects at varying heights.
- (3) Controlling the hand open/close function in concert with the wrist rotation function. They practiced tasks that involved grasping, transporting, rotating, and releasing objects at varying heights.

Following their practice period, participants were tested to determine whether they could reliably control the simulated prosthesis. Two cups were situated in front of them at two different heights, with a ball in one of the cups. Participants were asked to pour the ball between the two cups, and instances when they dropped the ball or a cup were recorded. If participants could not complete at least 10 pours with a success rate of at least 75% within 10 minutes, the session was ended, and they were removed from the study. Recall that one participant was removed (as stated in the Participants section), given that they could not complete this activity with LDA-Baseline in their first session.

#### 6.4.6. Motion Capture Setup & Kinematic Calibrations

For participants who were deemed eligible for controller testing, the following motion capture steps were undertaken:

**Step 1: Motion Capture Setup**—An 8-camera OptiTrack Flex 13 motion capture system (Natural Point, OR, USA) was used to capture participant movements and task objects at a sampling rate of 120 Hz. Eight individual markers were placed on the simulated prosthesis hand, circled in Figure 6-5 (one on the thumb, one on the index finger, and the remaining six throughout the back and side of the hand to ensure reliable rigid body tracking). Rigid marker plates were also placed on each

participant's right forearm (affixed to the simulated prosthesis socket), upper arm, and thorax, in accordance with Boser et al.'s cluster-based marker model [150].

**Step 2: Kinematic Calibrations**—Each participant was required to perform two kinematics calibrations. As per Boser et al., the first calibration called for participants to hold an anatomical pose [150], for capture of the relative positions of the hand markers and motion capture marker plates when wrist rotation and shoulder flexion/extension angles were at 0°. The second calibration required participants to hold a ski pose [150], for the purpose of refining wrist rotation angles. Here, three additional individual markers were affixed to the simulated prosthesis, as shown in Figure 6-5:

- (1) One marker placed on the top of the prosthesis' hand motor, with the device hand closed
- (2) One marker placed on the bottom of the prosthesis' wrist motor, forming a line with the first marker (to represent the axis about which the wrist rotation occurred)
- (3) One marker placed on the side of the prosthesis' wrist motor (to create a second axis, perpendicular to the axis of wrist rotation)

Upon completion of the two kinematics calibrations, all Step 2 markers were removed. What remained were only those markers affixed during Step 1 for data collection purposes.

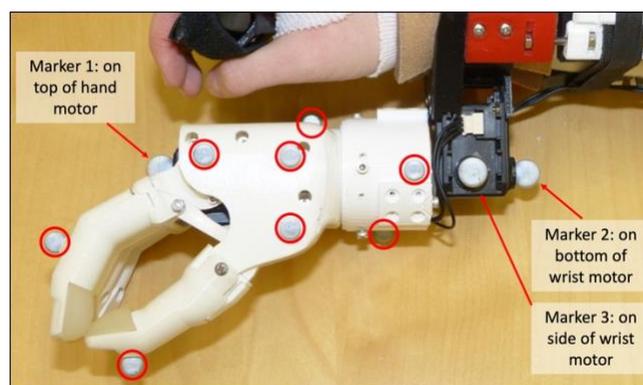


Figure 6-5. Motion capture markers affixed to the simulated prosthesis. The eight motion capture markers that remained attached to the hand are circled, and the three additional individual markers for the ski pose calibration are labelled.

#### 6.4.7. Functional Tasks & Data Collection

Motion capture data were collected during the execution of the following functional tasks:

**Pasta Box Task (Pasta)**—Participants were required to perform three distinct movements, where they transported a pasta box between a 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> location (a side table and two shelves at varying heights on a cart, including across their midline) [95]. The task setup is shown in Figure 6-6A. Motion capture markers were placed on all task objects, as per Valevicius et al. [95]. Participants performed a total of 10 Pasta trials. If participants dropped the pasta box, placed it incorrectly, performed an incorrect movement sequence, or hit the frame of the task cart, the trial was not analyzed. Pasta was the first of two functional tasks performed as it was considered easier.

**RCRT**—Participants were required to perform three distinct movements using clothespins. They moved three clothespins between 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> locations on horizontal and vertical bars [96]. To simplify trial execution, RCRT was split into RCRT Up and RCRT Down trials. The task setup for these trials is shown in Figure 6-6B. During Up trials, participants moved the three clothespins from the horizontal bar to the vertical bar, and during Down trials, they moved the clothespins from the vertical bar to the horizontal bar. A height adjustable cart was set such that the top of each participants' shoulder was aligned with the midpoint between the top two targets on the vertical bar. Motion capture markers were placed on all task objects, as per our earlier research [90]. Participants performed a total of 10 Up trials and 10 Down trials. If participants dropped a clothespin, placed it incorrectly, or performed an incorrect movement sequence, the trial was not analyzed. Performance of RCRT Up and Down trials were alternated, and started with RCRT Up.

#### 6.4.8. Survey Administration

At the end of each session, each participant completed two surveys: the NASA-TLX [145] and a usability survey [147]. The former was administered using the official **NASA-TLX** iPad application, where participants scored their device control workload demand on a continuous rating scale with endpoint anchors of low and high. The **usability survey** was administered on paper, where participants marked their usability scores on a continuous rating scale with endpoint anchors of 0 and 5. In their second session, participants were *not* reminded of their survey responses from their first session.

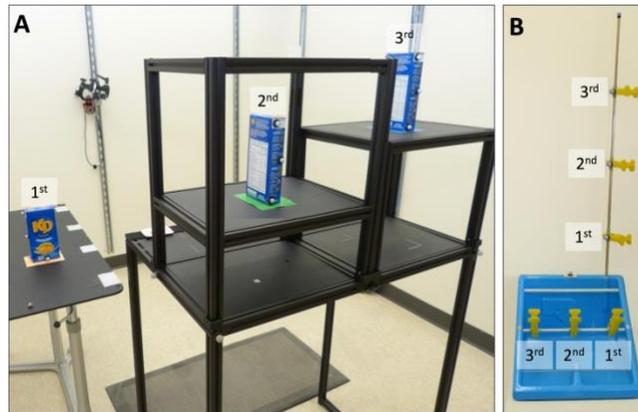


Figure 6-6. Task setup for (A) Pasta and (B) RCRT Up and Down. In panel (A), the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> pasta box locations are labelled. The pasta box movement sequence is 1<sup>st</sup> → 2<sup>nd</sup> → 3<sup>rd</sup> → 1<sup>st</sup> locations. In panel (B), the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> clothespin locations on the horizontal and vertical bars are labelled. The clothespin movement sequences in RCRT Up are horizontal 1<sup>st</sup> → vertical 1<sup>st</sup>, horizontal 2<sup>nd</sup> → vertical 2<sup>nd</sup>, and horizontal 3<sup>rd</sup> → vertical 3<sup>rd</sup> locations. The clothespin movement sequence in RCRT Down follows the same order, but with each clothespin moved from vertical to horizontal locations.

#### 6.4.9. Data Processing & Calculation Procedures

**Motion Capture Data Cleaning & Calculations**—Motion capture marker trajectory data were cleaned and filtered. As per Valevicius et al. [95], grip aperture was calculated as the distance between the motion capture markers on the simulated prosthesis' index and thumb, and a 3D object representing the simulated prosthesis' hand was generated using the remaining 6 hand motion capture markers. Then, through calculations modified from Boser et al. [150], wrist rotation was calculated using the forearm and hand motion capture markers, and shoulder flexion/extension was calculated using the upper arm and thorax motion capture markers.

**Data Segmentation**—The task data were segmented in accordance with Valevicius et al. [95], as follows:

- For each *task*, the data from each *trial* were first divided into distinct *movements 1, 2, and 3* based on hand velocity and the velocity of the pasta box/clothespins during transport.

**Pasta** Movements 1, 2, and 3 differentiated between: (1) reaching for the pasta box at its 1<sup>st</sup> location, grasping it, transporting it to its 2<sup>nd</sup> location, releasing it, and moving their hand back to a home position; (2) reaching for the pasta box at the 2<sup>nd</sup> location, grasping it, transporting it to its 3<sup>rd</sup> location, releasing it, and moving their hand back to the home

position; and (3) reaching for the pasta box at the 3<sup>rd</sup> location, grasping it, transporting it back to the 1<sup>st</sup> location, releasing it, and moving their hand back to the home position.

**RCRT Up** Movements 1, 2, and 3 differentiated between: (1) reaching for the 1<sup>st</sup> clothespin at its 1<sup>st</sup> horizontal location, grasping it, transporting it to its 1<sup>st</sup> vertical location, releasing it, and moving their hand back to a home position; (2) reaching for the 2<sup>nd</sup> clothespin at its 2<sup>nd</sup> horizontal location, grasping it, transporting it to its 2<sup>nd</sup> vertical location, releasing it, and moving their hand back to the home position; and (3) reaching for the 3<sup>rd</sup> clothespin at its 3<sup>rd</sup> horizontal location, grasping it, transporting it to its 3<sup>rd</sup> vertical location, releasing it, and moving their hand back to the home position.

**RCRT Down** Movements 1, 2, and 3 differentiated between: (1) reaching for the 1<sup>st</sup> clothespin at its 1<sup>st</sup> vertical location, grasping it, transporting it to its 1<sup>st</sup> horizontal location, releasing it, and moving their hand back to a home position; (2) reaching for the 2<sup>nd</sup> clothespin at its 2<sup>nd</sup> vertical location, grasping it, transporting it to its 2<sup>nd</sup> horizontal location, releasing it, and moving their hand back to the home position; and (3) reaching for the 3<sup>rd</sup> clothespin at its 3<sup>rd</sup> vertical location, grasping it, transporting it to its 3<sup>rd</sup> horizontal location, releasing it, and moving their hand back to the home position.

- Then, the data from each of the three movements were further segmented into *five phases* of (1) Reach, (2) Grasp, (3) Transport, (4) Release, and (5) Home (note that the Home phase was not used for data analysis)
- Finally, *two movement segments* of (1) Reach-Grasp and (2) Transport-Release were defined for select metrics analysis
- Six final levels for data analysis resulted: *controller* (either RCNN-TL or LDA-Baseline), *task* (either Pasta, RCRT Up, or RCRT Down), *trial* (1–10), *movement* (1–3), *movement segment* (Reach-Grasp or Transport-Release), and *phase* (Reach, Grasp, Transport, or Release)

**Grip Aperture & Wrist Rotation Re-Calculations**—The grip aperture and wrist rotation angle were re-calculated using the data from the simulated prosthesis’ two motors, given that small (yet informative) adjustments in the positions of these motors may not have been detected by motion capture cameras. The positions of these motors were first upsampled to 120 Hz using linear

interpolation. *Grip aperture re-calculation*: motion-capture-calculated grip aperture was used to fit a trinomial curve to transform the hand motor data to grip aperture. *Wrist motor angle re-calculation*: motion-capture-calculated wrist rotation was used to fit a binomial curve to transform the wrist motor data to wrist rotation angles.

**Final Suite of Metrics Calculations**—The final suite of metrics was calculated using the procedures described in Table 6-1, Table 6-2, and Table 6-3. Note that the simultaneous wrist-shoulder movements metric was calculated only for Reach and Transport phases of RCRT Up and RCRT Down trials, because these were the only phases that required the participant to rotate the device wrist while moving their arm to a different height.

#### 6.4.10. Statistical Analysis

To investigate task performance difference between RCNN-TL and LDA-Baseline, the following statistical analyses were performed:

**For metrics that were analyzed at the phase or movement segment level**—Participants' results were first averaged across trials and movements. If results were normally-distributed, a two-factor repeated-measures analysis of variance (RMANOVA) was conducted using the factors of controller and phase/movement segment. When the resulting controller effects or controller-phase/movement segment interactions were deemed significant (that is, when the Greenhouse-Geisser corrected p value was less than 0.05), pairwise comparisons between the controllers were conducted. If results were not normally-distributed, the Friedman test was conducted. When the resulting p value was less than 0.05, pairwise comparisons between the controllers were conducted. Pairwise comparisons (t-test/Wilcoxon sign rank test) were deemed significant if the p value was less than 0.05.

**For metrics that were analyzed at the trial level**—Participants' results were first averaged across trials, then pairwise comparisons were conducted as detailed above.

**For metrics that were analyzed at the task or controller level**—Pairwise comparisons were conducted as detailed above.

### 6.4.11. Limb Position Effect Identification

The limb position effect has been shown to cause control accuracy degradation and large between-participant control variation in offline research [123]. However, earlier works have not pinpointed specific instances of the effect in functional task execution data. Using the novel control characteristics metrics introduced in this work, identification of such instances is possible—larger medians and/or larger interquartile ranges (IQRs) can provide evidence of degraded control. To identify the limb position effect in this study, metrics’ medians and IQRs for Reach, Grasp, Transport, and Release phases were considered separately across movements 1, 2, and 3 of Pasta, RCRT Up and RCRT Down. An occurrence where movement variation was *not* due to the limb position effect is illustrated in Figure 6-7A, where the number of wrist rotation adjustments metric in RCRT Down Release phases have medians and IQRs that remain relatively constant at different limb positions. Conversely, an occurrence where movement variation *was* due to the limb position effect can be seen in Figure 6-7B. Here, the same metric in RCRT Down Grasp phases shows its medians and IQRs both increasing as the limb position changed.

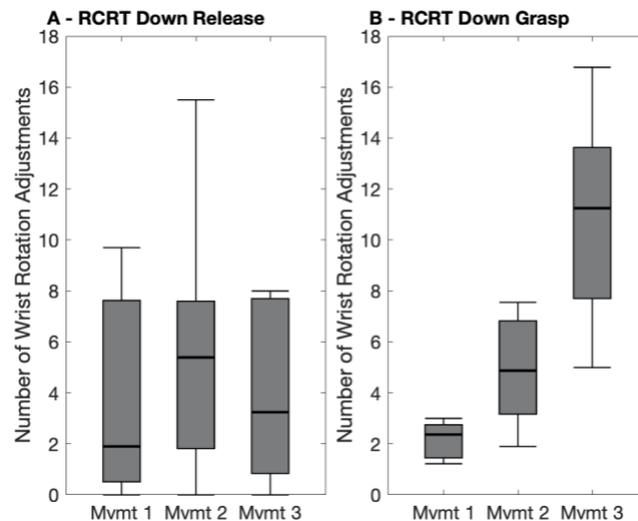


Figure 6-7. Box plots indicating LDA-Baseline number of wrist rotation adjustments in each (A) RCRT Down Release and (B) RCRT Down Grasp of each task movement (Mvmt). Medians are indicated with thick lines, and interquartile ranges are indicated with boxes.

The following limb position identification process was used to examine all control characteristics metrics for Reach, Grasp, Transport, and Release phases across the three movements of Pasta, RCRT Up, and RCRT Down:

- First, the three medians were rescaled as percentages of the maximum of the three medians
- Next, the three IQRs were rescaled as percentages of the maximum of the three IQRs
- Then, the limb position effect identification rules outlined in Table 6-4 were developed—through iterative trial-and-error comparisons of potential rules to visual representations of metrics’ medians and IQRs (as in Figure 6-7A, B)
- The resulting rules were subsequently used to identify instances of the limb position effect. Note that for Pasta, two rule options were used, given that the limb position effect was most likely to be present in that task’s movements 2 or 3 (at the highest shelf location). For RCRT Up and RCRT Down, only one rule option was necessary, given that the limb position effect was most likely to occur in movement 3 (at the top clothespin location). For each rule, the limb position effect was identified only in instances where all conditions were met (that is, when all rules in a row of Table 6-4 were true).

Table 6-4. Limb position effect identification rules for control characteristics metrics. The rules used for limb position effect identification are based on each movement’s median and interquartile range (IQR). In each row, all conditions had to be true for a positive identification of the limb position effect.

<b>Limb Position Effect Identification Rule</b>	<b>Conditions Identifying Limb Position Effect</b>					
	<b>Movement 1 Median</b>	<b>Movement 2 Median</b>	<b>Movement 3 Median</b>	<b>Movement 1 IQR</b>	<b>Movement 2 IQR</b>	<b>Movement 3 IQR</b>
Pasta Reach and Grasp Rule	< 90%	< 75%	= 100%		< 55%	= 100%
Pasta Transport and Release Rule		= 100%	< 85%		= 100%	
RCRT Up and RCRT Down Rule	< 55%	< 85%	= 100%	< 25%		> 80 %

The abovementioned rules are applicable to metrics where smaller values are indicative of control improvements, as was the case with most control characteristics metrics in this work. The exception was the simultaneous wrist-shoulder movements metric, where *larger* values were indicative of improved control. To adjust for this exception, each of the three movements' rescaled medians (represented as percentages) were subtracted from 100%. In doing so, these medians were altered to represent the percent of the phase in which simultaneous movements did *not* occur. After this adjustment, the limb position identification process could be followed.

## 6.5. Results

### 6.5.1. Task Performance

The significant differences across the task performance metrics are reported in Table 6-5. Task specific outcomes derived from the table include:

**Pasta**—RCNN-TL performed significantly better than LDA-Baseline in *one* metric: Release phase duration.

**RCRT Up**—LDA-Baseline performed significantly better than RCNN-TL in *one* metric: success rate.

**RCRT Down**—RCNN-TL performed significantly better than LDA-Baseline in *one* metric: Grasp relative phase duration. LDA-Baseline performed significantly better than RCNN-TL in *one* metric: release relative phase duration.

**Summary**—Only 4 of the 48 Task Performance metrics showed significant differences, 2 of which demonstrated that RCNN-TL performed better than LDA-Baseline. It appears that a richer set of metrics is needed to better understand such outcomes—beyond those derived from task performance metrics alone.

### 6.5.2. Control Characteristics

The significant differences across control characteristics metrics are reported in Table 6-6. Task specific outcomes derived from the table include:

**Pasta**—RCNN-TL performed significantly better than LDA-Baseline in *seven* metrics. Examples of one such metric are illustrated in Appendix C (for Reach-Grasp grip aperture plateau). The limb position effect was identified in *12* metrics under LDA-Baseline control, and in one metric (Grasp total muscle activity) under RCNN-TL control. Four of the seven significant RCNN-TL versus LDA-Baseline differences were in metrics that showed evidence of the limb position effect.

**RCRT Up**—No significant differences were identified, and no metrics showed evidence of the limb position effect.

**RCRT Down**—RCNN-TL performed significantly better than LDA-Baseline in *two* metrics. Examples of one such metric are illustrated in Appendix C (for Reach simultaneous wrist-shoulder movements). The limb position effect was identified in *four* other metrics, with one such instance illustrated in Figure 6-7B (Grasp number of wrist rotation adjustments).

**Summary**—9 of the 81 Control Characteristics metrics showed significant differences, all of which demonstrated that RCNN-TL performed better than LDA-Baseline. Furthermore, 16 metrics showed evidence of the limb position effect. All such metrics were only identified in Pasta and RCRT Down, suggesting that these outcomes are likely influenced by the position-aware nature of RCNN-TL control.

Table 6-5. Task Performance metrics results. Each cell contains the RCNN-TL median (and interquartile range in parentheses) in the first line and the LDA-Baseline median (interquartile range) in the second line. Green cells indicate metrics in which RCNN-TL performed significantly better than LDA-Baseline (solid green:  $p < 0.005$ , dense green grid:  $p < 0.01$ ). Red cells indicate metrics in which LDA-Baseline performed significantly better than RCNN-TL (sparse red grid:  $p < 0.05$ ).

Task		Pasta	RCRT Up	RCRT Down
Success Rate (%)		95.00 (15.00)	75.00 (35.00)	100.00 (10.00)
		95.00 (10.00)	100.00 (5.00)	90.00 (20.00)
Trial Duration (s)		25.28 (9.86)	31.62 (8.14)	31.12 (10.37)
		30.78 (16.28)	27.85 (13.40)	31.72 (6.87)
Phase Duration (s)	Reach	1.28 (0.75)	1.61 (0.88)	1.40 (1.21)
		1.38 (1.03)	1.52 (1.17)	1.35 (0.93)
	Grasp	1.59 (1.46)	2.09 (1.54)	1.67 (1.33)
		1.84 (1.33)	1.68 (1.31)	2.28 (3.19)
Transport	2.40 (0.69)	2.90 (1.24)	2.56 (0.72)	
	2.54 (1.04)	2.89 (2.04)	2.59 (1.44)	
Release	0.88 (0.62)	1.96 (1.28)	2.19 (0.93)	
	1.11 (1.31)	1.48 (0.87)	1.57 (1.28)	
Relative Phase Duration (%)	Reach	23.08 (4.81)	18.62 (6.49)	18.03 (8.17)
		20.80 (8.48)	19.49 (7.34)	15.83 (4.65)
	Grasp	24.95 (10.10)	23.68 (10.46)	22.77 (8.39)
		23.65 (10.34)	23.42 (8.01)	27.45 (14.61)
Transport	38.07 (9.53)	33.62 (10.38)	32.15 (11.38)	
	32.81 (11.95)	35.82 (7.46)	30.44 (10.80)	
Release	16.45 (9.30)	20.82 (13.73)	25.80 (7.39)	
	18.57 (10.91)	19.75 (6.75)	20.23 (8.73)	
Peak Hand Velocity (mm/s)	Reach-Grasp	780.06 (265.86)	559.84 (243.55)	855.12 (425.89)
		801.79 (291.61)	543.16 (254.90)	837.28 (322.24)
Transport-Release		768.74 (387.29)	486.19 (272.47)	485.99 (281.39)
		789.18 (424.73)	495.46 (286.86)	444.93 (314.93)
Hand Distance Travelled (mm)	Reach-Grasp	689.38 (298.01)	537.06 (178.13)	679.54 (251.59)
		766.21 (379.84)	464.67 (192.90)	684.68 (543.83)
Transport-Release		931.14 (420.62)	610.88 (324.93)	513.42 (303.34)
		978.56 (396.90)	584.27 (321.36)	512.70 (365.95)
Hand Trajectory Variability (mm)	Reach-Grasp	44.37 (42.81)	35.74 (21.68)	50.11 (32.74)
		68.58 (51.60)	32.56 (18.68)	70.92 (46.82)
Transport-Release		53.92 (28.02)	39.32 (31.70)	33.74 (28.57)
		82.75 (52.25)	40.83 (37.77)	42.02 (32.92)

Table 6-6. Control Characteristics metrics results. Each cell contains the RCNN-TL median (and interquartile range in parentheses) in the first line and the LDA-Baseline median (interquartile range) in the second line. Green cells indicate metrics in which RCNN-TL performed significantly better than LDA-Baseline (solid green:  $p < 0.005$ , dense green grid:  $p < 0.01$ , sparse green grid:  $p < 0.05$ ). Dark grey cells indicate instances in which a metric was not relevant. Dark cell borders indicate metrics that displayed evidence of the limb position effect under LDA-Baseline control. A double cell border indicates the metric that displayed evidence of the limb position effect under both LDA-Baseline and RCNN-TL control. (Table 6-6 found on next page)

Task		Pasta	RCRT Up	RCRT Down
Grip Aperture Total Movement (mm)	Reach	10.37 (13.46) 4.77 (12.80)	20.16 (41.19) 18.53 (30.34)	10.12 (23.23) 11.87 (21.75)
	Grasp	66.02 (33.61) 69.27 (33.10)	77.88 (51.67) 61.22 (29.51)	72.50 (20.96) 75.99 (58.61)
	Transport	18.55 (7.78) 13.54 (15.50)	13.05 (8.13) 10.51 (13.76)	11.88 (8.36) 10.28 (7.56)
	Release	51.50 (15.89) 56.90 (13.77)	60.06 (19.01) 57.85 (9.42)	60.64 (9.82) 56.49 (10.75)
Wrist Rotation Total Movement (°)	Reach	7.57 (14.51) 4.92 (15.80)	60.46 (55.57) 41.64 (32.46)	32.39 (44.86) 16.04 (30.04)
	Grasp	14.89 (21.38) 25.32 (37.22)	22.87 (39.61) 14.74 (25.36)	17.82 (20.64) 27.90 (80.36)
	Transport	13.13 (24.25) 26.55 (45.64)	73.90 (55.64) 64.23 (53.11)	66.91 (32.28) 50.78 (49.08)
	Release	2.92 (9.78) 20.38 (54.62)	20.91 (42.18) 8.05 (18.85)	31.22 (25.14) 18.35 (54.44)
Number of Grip Aperture Adjustments	Reach	0.44 (1.31) 0.37 (1.56)	2.33 (3.10) 2.11 (2.49)	1.00 (2.21) 0.89 (1.94)
	Grasp	2.98 (4.00) 3.74 (4.46)	3.50 (4.81) 3.20 (4.69)	3.30 (1.79) 5.38 (8.88)
	Transport	2.17 (1.68) 2.84 (2.40)	3.52 (2.33) 3.40 (3.41)	3.05 (2.40) 2.90 (2.47)
	Release	1.78 (1.97) 2.56 (3.84)	3.80 (2.74) 2.60 (2.81)	4.11 (2.85) 3.11 (3.40)
Number of Wrist Rotation Adjustments	Reach	0.68 (2.18) 0.30 (1.84)	2.32 (2.66) 2.06 (2.60)	2.50 (2.15) 1.44 (2.56)
	Grasp	2.65 (4.65) 4.72 (6.44)	2.55 (2.04) 2.05 (3.55)	2.84 (3.38) 4.47 (9.95)
	Transport	2.40 (4.29) 4.43 (6.00)	4.69 (3.38) 4.20 (3.33)	2.90 (2.78) 3.94 (4.96)
	Release	0.68 (2.09) 3.18 (5.89)	3.10 (5.48) 1.90 (3.44)	3.35 (3.40) 3.24 (6.88)
Grip Aperture Plateau (s)	Reach-Grasp	1.49 (0.84) 1.91 (1.39)	1.69 (0.61) 1.66 (0.81)	1.79 (1.02) 1.66 (0.89)
Simultaneous Wrist-Shoulder Movements (%)	Reach		27.36 (16.02) 31.40 (17.10)	23.77 (29.15) 9.70 (13.46)
	Transport		28.68 (13.48) 22.25 (20.74)	17.97 (14.10) 10.57 (12.61)
Total Muscle Activity	Reach	87.60 (49.47) 82.38 (100.06)	159.53 (252.54) 184.27 (137.85)	130.99 (155.05) 90.73 (102.13)
	Grasp	152.98 (168.33) 186.28 (240.38)	232.44 (220.79) 153.28 (115.47)	138.10 (150.30) 190.20 (327.96)
	Transport	181.99 (88.61) 183.95 (114.74)	244.02 (133.68) 208.27 (139.09)	226.15 (138.11) 248.58 (84.72)
	Release	85.77 (107.96) 120.56 (150.44)	207.73 (157.52) 148.54 (96.00)	226.28 (161.05) 184.11 (161.54)

### 6.5.3. User Experience

User experience metrics were calculated at the controller level, rather than for each task (detailed in Table 6-3). There were no significant differences between RCNN-TL and LDA-Baseline. Box plots illustrating the median controller-level scores across participants can be found in Figure 6-8.

Of note, RCNN-TL scored better than LDA-Baseline in the NASA-TLX's Mental Demand dimension and in the usability survey's Intuitiveness dimension. These results suggest that RCNN-TL offered more intuitive control. The two controllers had equal median scores in NASA-TLX's Effort dimension. LDA-Baseline scored better than RCNN-TL in all other dimensions.

## 6.6. Discussion

The suite of myoelectric prosthesis control evaluation metrics introduced in this work (detailed in Table 6-1, Table 6-2, and Table 6-3) yielded informative limb position effect-related outcomes that could only be speculated upon in our earlier work [90]. What follows is a discussion about the metrics-driven findings from this current work, with a focus on understanding *when* and *why* limb position variations caused control challenges during participants' execution of the Pasta Box Task (Pasta) and the Refined Clothespin Relocation Test (RCRT Up and RCRT Down).

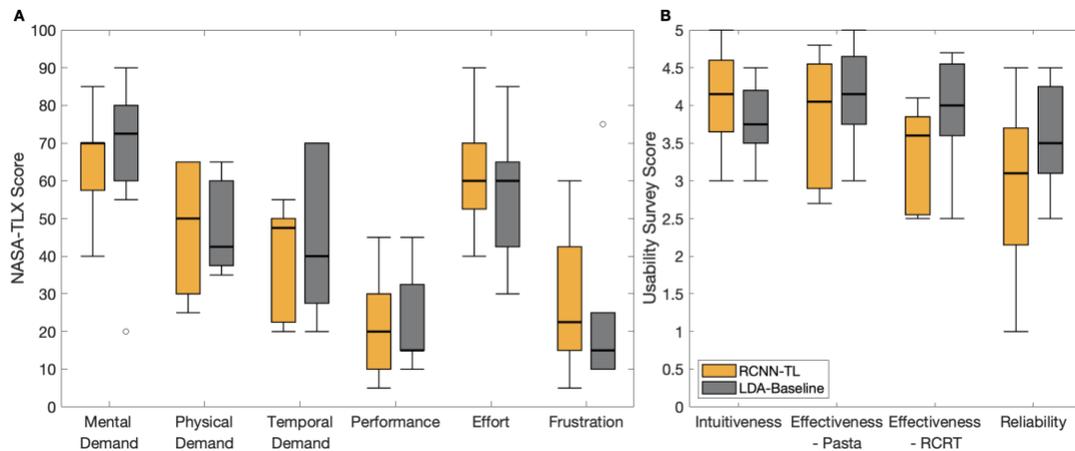


Figure 6-8. Box plots indicating user experience metrics results with RCNN-TL (orange) and LDA-Baseline (grey) for: (A) NASA-TLX, and (B) usability survey. Medians are indicated with thick lines, interquartile ranges are indicated with boxes, and outliers are indicated with circles.

### 6.6.1. Findings from Control Characteristics Metrics

**Limb Position Effect Identification**—To our knowledge, no other study has identified occurrences of the limb position effect using functional task assessment outcomes. In this work, we used functional tasks to assess device control and found that analysis of our control characteristics metrics *did* uncover instances of the limb position effect. In LDA-Baseline results, instances were identified in Pasta and RCRT Down (16 of 81 cells with dark borders in Table 6-6), and never in RCRT Up. Note that for RCNN-TL, one instance of the limb position effect was uncovered—Grasp total muscle activity in Pasta. However, this instance may simply be due to inevitable positional EMG signal variations, rather than due to control degradation. Based *solely on LDA-Baseline results*, we surmised the following:

- Raised arm positions in the sagittal plane caused grasp challenges for participants, as evidenced by the identification of the limb position effect only during the Grasp phases of RCRT Down (in four metrics).
- Raised arm positions in the sagittal plane did *not* cause release challenges for participants, as evidenced by the *absence* of limb position effect identification in RCRT Up. Logically, as hand opening during object release phases is controlled by wrist extension muscle activation, classification of wrist extension was not affected by the limb position effect.
- Arm movements along the frontal plane caused further control challenges for participants. Not only did Pasta require participants to perform arm raises in the sagittal plane, large cross-body and away-from-body movements had to be introduced to accomplish this task. The limb position effect was detected in three of four Pasta phases (four times in Reach, once in Reach-Grasp, three times in Grasp, and four times in Release).

So overall, LDA-Baseline control often appeared to be impeded by shoulder position fluctuations in the frontal plane. Furthermore, arm raises limited to the sagittal plane caused only grasp control deterioration for this controller. Both such circumstances identify catalysts for limb position effect control challenges.

**Evidence of Limb Position Effect Mitigation**—Recall that Table 6-6 also identified significant differences in control characteristics metrics, where green cells indicated instances where RCNN-

TL performed significantly better than LDA-Baseline. Nine of the 81 metrics (cells) were significant (shaded in green), indicating that RCNN-TL always performed the same as, or significantly better than, LDA-Baseline for these metrics. Furthermore, all such significant differences presented in Table 6-6 occurred in Pasta and RCRT Down, and never in RCRT Up. This coincides with those tasks where instances of the limb position effect were identified, suggesting that RCNN-TL successfully mitigated such occurrences.

Interestingly, RCNN-TL performed significantly better than LDA-Baseline in several Pasta metrics, even though Pasta involves numerous limb positions that were not included in RCNN-TL's pre-training/retraining routines. We speculate that the pre-training data from 19 individuals provided sufficient variety to result in a controller that is robust to limb positions not included in its training routines.

Significant control characteristics differences between RCNN-TL and LDA-Baseline were not identified for RCRT Up. Recall that the limb position effect was *not* identified in RCRT Up, despite this task's requirement for varied limb positions. So, if the problem did not cause control degradation for either controller, then perhaps: (a) LDA-Baseline simply performed well during this task and control improvements were not necessary, or (b) RCNN-TL control should be improved in instances when the limb position effect is *not* evident.

**Merits of Control Characteristics Metrics**—Significant differences between RCNN-TL and LDA-Baseline were identified in at least one phase/movement segment for all control characteristics metrics analyzed, with the exceptions of total grip aperture movement and total muscle activity. Still, these two metric exceptions might yield outcomes beneficial to other controller comparisons and should not be discounted from the metrics introduced in this work. Total grip aperture movement, for instance, might help to identify grasping efficiency during task execution, and total muscle activity might help to identify muscle exertion required for task completion. Future controller comparisons are expected to determine whether these metrics are sensitive to controller variations.

### 6.6.2. Findings from Task Performance Metrics

Table 6-5 identified 2 of 48 metrics that showed RCNN-TL performing significantly better than LDA-Baseline (for Release phase duration in Pasta and Grasp relative phase duration in RCRT

Down), and 2 of 48 metrics that showed the contrary (for success rate in RCRT Up and Release relative phase duration in RCRT Down). These outcomes coincided with those of our earlier work [90], however, the control characteristics metrics introduced in this study facilitated a deeper understanding of why task performance deteriorated at times—specifically, when instances of the limb position effect hampered control. The following task performance insights were uncovered in this work:

- RCNN-TL successfully mitigated the limb position effect, as evidenced by the two specific instances when its control was significantly better than that of LDA-Baseline—in Pasta Release and RCRT Down Grasp phases. Our control characteristics analysis revealed that participants struggled during these phases (identified as instances where the limb position effect occurred). Such struggles were apparent when participants used LDA-Baseline, but not so when using RCNN-TL. So, RCNN-TL likely remedied control degradation introduced by the limb position effect.
- RCNN-TL may not have performed well in instances where the limb position effect was *not* evidenced. Consider that two task performance metrics showed that RCNN-TL performed significantly *worse* than LDA-Baseline: (1) the relative duration of the RCRT Down Release phases, and (2) the RCRT Up success rate. Furthermore, consider that analysis of control characteristics revealed that device control was *not* affected by the limb position effect in either the RCRT Down Release phase, or any phase of RCRT Up. Together, these outcomes support the hypothesis that RCNN-TL may not have performed well in instances where the limb position effect was *not* evidenced.
  - Of note, the lower RCRT Up success rate was due to clothespins being dropped by participants. This tendency towards unintended hand opening is in keeping with RCNN-TL’s control characteristics results versus those of LDA-Baseline, with higher medians in grip aperture total movement, higher medians in number of grip aperture adjustments, along with larger IQRs in Reach and Grasp phases of these same metrics.

### 6.6.3. Findings from User Experience Metrics

Figure 6-8 presented the user experience metrics. Despite this work's improved control outcomes, no significant differences were identified in the NASA-TLX and usability surveys. This work uncovered the following insights, to guide future use of user experience metrics:

- Without the provision of participants' scores from their first testing session upon return for their second session (following the washout period of at least one week), their initial anchor scores of "good" and "poor" were not likely to have been precisely recalled.
- Given that participants were without limb loss, they only had a perception of fully functional control using their intact hand and wrist. They did not have a baseline perception of poor or diminished control, as none had prior experience with a simulated prosthesis. As a result, their subjective anchor scores of "good" and "poor" in their first session were likely influenced by their perception of perfect control, whereas in their second session, they might have been further influenced by their first session's simulated prosthesis control experience.

Overall, a specific question about which controller each participant preferred (asked at the end of their second session) would better gauge their controller partiality. In addition, reminding participants of their first-session scores immediately prior to their second session survey completion, might address expectation-related variability in anchor scores [151]. After this current study was conducted, a new Prosthesis Task Load Index (PROS-TLX) was developed and validated [152], and should be considered in future comparative prosthesis control research.

### 6.6.4. Evidence of Training Routine Reduction

The General Participant Group performed a long (200-second) pre-training routine prior to RCNN-TL use. This pre-training duration is similar to that of position-aware controller solutions in the literature [15], [16], [19], [20], [21], [22], [23]. RCNN-TL retraining, as performed by the SP Participant Group, was accomplished using a shortened (60-second) routine. That is, a 70% decrease in model training duration resulted due to the introduction of transfer learning. This current research, therefore, confirms that TL *is* a valuable adjunct to RCNN-based classification control, as it offers a model starting point that needs only to be calibrated using a smaller amount

of individual-specific data. Notably, a TL solution is not possible with LDA-based control. Overall, this training routine reduction solution shows promise towards solving the limb position effect challenge—*without* the requirement of a burdensome training routine.

### 6.6.5. Limitations

As a first limitation of this study, participants without upper limb impairment were recruited rather than myoelectric prosthesis users. Although these participants learned how to control a simulated prosthesis, further practice may have been necessary to accurately represent the control capabilities of myoelectric prosthesis users. Secondly, the implementation of the surveys may not have adequately captured user experience data. Thirdly, previous work had suggested that the conditions under which the RCNN-TL pre-training data was collected could result in control flaws [90]—because pre-training data from participants *without* a donned simulated prosthesis were too dissimilar from those *with* a donned simulated prosthesis (the conditions under which the retraining and device use occurred). This testing condition dissimilarity consideration was not examined in the current study. Finally, although our experimentation used optical motion capture technology to gather movement data, the use of this costly equipment might not always be feasible. As such, our suite of control metrics is not solely reliant on motion capture data for metrics calculations. Alternative sources that capture grip aperture and wrist rotation data (such as from device motor positions), along with shoulder flexion/extension data (such as from markerless motion capture technology or IMUs [153]) can be used for these calculations. Furthermore, other methods of segmenting functional tasks into Reach, Grasp, Transport, and Release phases (such as segmenting markerless motion capture data or IMU-captured data) can be implemented in preparation for metrics calculations.

### 6.6.6. Future Work

**RCNN-TL future work**—Next steps for RCNN-TL should focus on improvements to device control in instances when the limb position effect is least likely to occur. Improvements to pre-training data collection conditions need to be studied. Examination of RCNN-TL using myoelectric prosthesis users is also a necessary step.

**Suite of control evaluation metrics future work**—To verify whether the suite of metrics introduced in this study are beneficial, future work should examine different controllers, using

both participants without limb loss wearing a simulated prosthesis and actual myoelectric prosthesis users operating a commercial device. Further use of the suite of metrics in control research is expected to uncover recommended subsets of discriminant metrics, based on each study's goals and experimentation methods—for instance, studies with tasks that do not require wrist rotation need not examine simultaneous wrist-shoulder movements; they should instead focus on metrics such as the number of grip aperture adjustments and grip aperture plateau time.

## 6.7. Conclusion

This work reinvestigated earlier comparative RCNN-TL versus LDA-Baseline research, which recommended that pattern recognition-based control not be judged by *task performance* alone, but rather, that *control characteristics* also be measured [90]. Then collectively, the task performance and control characteristics should be weighed against qualitative *user experience* [90]. The current study heeded these recommendations, and in doing so, contributed and tested a viable suite of myoelectric prosthesis control evaluation metrics for use in future comparative control model research. Using these metrics, this study has contributed insights into occurrences and implications of the limb position effect challenge and offered validation that TL-based neural network control solutions show promise towards solving this pervasive problem. The suite of metrics introduced and subsequently used in this work is expected to benefit future research intent on improving rehabilitation device control.

# Chapter 7.

## A Comparison of RCNN Classification and Regression Control

Chapter 7 presents “Myoelectric Prosthesis Control using Recurrent Convolutional Neural Network Regression Mitigates the Limb Position Effect,” originally submitted to Neural Computing and Applications, in 2024 [154]. It addresses the research gaps identified in Chapter 3 and Chapter 6. In this current chapter, the novel recurrent convolutional neural network classification- and regression-based control solutions presented in Chapter 3 are again tested, with their resulting control assessed through an analysis of the suite of metrics introduced in Chapter 6. Notably, participants *without* amputation wore a simulated prosthesis during such testing. The promising outcomes of this work suggest that control testing using persons *with* transradial amputation is warranted. An illustration of how the information presented in Chapter 7 contributes to the overall work in this thesis is presented in Figure 7-1.

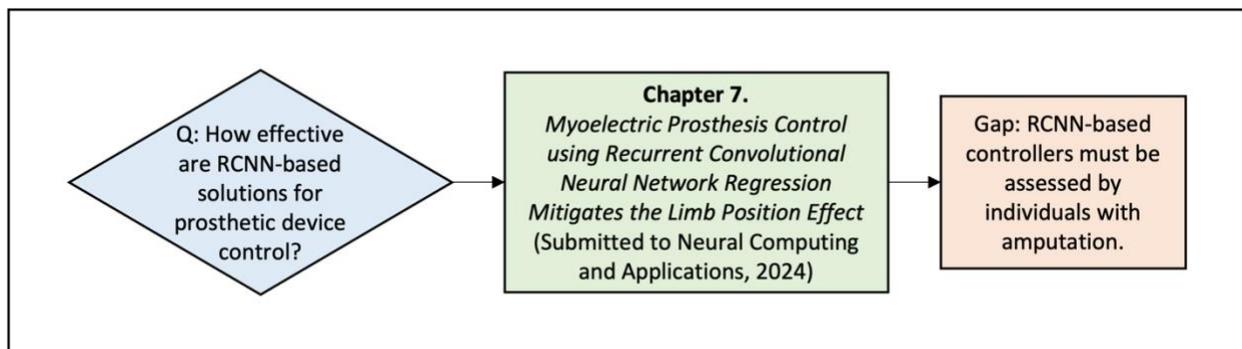


Figure 7-1. Research question addressed by Chapter 7 and remaining gap.

## Abstract

Although state-of-the-art myoelectric prostheses offer persons with upper limb amputation extensive movement capabilities, users have not been afforded a reliable means to control common movements required in daily living. Many proposed prosthesis controllers use pattern recognition, a method that learns and recognizes patterns of electromyographic (EMG) signals produced by the user's residual limb muscles to predict and execute device movements. Such control becomes unreliable in high limb positions—a problem known as the limb position effect. Pattern recognition often uses a classification algorithm; simple to implement, but limits user-initiated control to only one device movement at a time, at a fixed speed. To combat position-related control deficiencies and classification controller constraints, we developed and tested two recurrent convolutional neural network (RCNN) pattern recognition-based solutions: (1) an RCNN classification controller that uses EMG plus positional inertial measurement unit (IMU) signals to offer one-speed, sequential movement control; and (2) an RCNN regression controller that uses the same data capture technique to offer simultaneous control of multiple movements and device movement velocity. We assessed both RCNN controllers by comparing them to a commonly used linear discriminant analysis classification controller (LDA-Baseline). Participants without upper limb impairment were recruited to perform multipositional tasks while wearing a simulated prosthesis. Both RCNN classification and regression controllers showed improved functional task performance over LDA-Baseline, in 11 and 38 out of 125 metrics, respectively. This work contributes an RCNN regression-based controller that provides movements that are accurate, simultaneous, and proportional to EMG signal intensity, with all such capabilities amenable to other EMG-based technologies including prostheses, exoskeletons, and even muscle-activated video games.

## 7.1. Introduction

Despite the functional capabilities offered by state-of-the-art myoelectric prostheses for use by those with transradial (below elbow) amputation, users report that these modern devices are challenging to control and offer unnatural movement qualities [12], [155]. In response to user and clinical feedback, ongoing research aims to improve overall device usability, with an emphasis on control reliability [31]. Pattern recognition is a control method that has garnered much focus in

upper limb prosthesis research [6]. With such control, electromyographic (EMG) signals generated by the residual limb musculature of a user are captured by device socket electrodes, and then interpreted by a controller to predict and execute the user's intended movements. Simply stated, a prosthetic limb moves in response to muscles deliberately contracted by its user, as coordinated by a control algorithm.

Most advanced pattern recognition-based controllers tend to use a *classification* algorithm [30], which predicts one device action (or class) at a time [6]. The resulting prosthetic limb movements, consequently, appear sequential (robot-like) and are delivered at a pre-set speed—for instance, a device that offers wrist and hand capabilities cannot inherently move these components simultaneously or with varied velocity. Another control alternative uses a *regression* algorithm. This approach can predict multiple device movements at once, each of which are proportional to muscle contraction intensity. The outcome from this approach is that users can control their prosthetic wrist and hand more intuitively and in a smooth manner using varied velocities throughout reaching and grasping actions [6].

Whether a classification or regression algorithm (known as a model) is used for control, pattern recognition-based prosthesis movement predictions are contingent on earlier-captured movement data. Prior to device use, a prosthesis user must perform a series of predetermined muscle contractions using their residual limb—known as a training routine [7]. Through training, patterns observed in the captured muscle signals are programmatically associated with corresponding device actions. A training routine should not take an excessive amount of time, particularly if it must be executed to recalibrate control during prosthesis use. Indeed, user feedback surveys report that above all else (even above improved control), upper limb prosthesis users want quick-setup control solutions [156], [157].

Classification control is often employed in prosthesis research for a number of reasons: (1) it can reliably predict discrete hand and wrist classes, which reduces model complexity [31]; (2) its simpler model form can yield fewer errors versus complex ones [10], [31]; and (3) its model training routines only require execution of static muscle contractions [158], making them straightforward to develop and quick to administer. Still, predicted movement errors are known to occur with classification control, particularly during transitions between hand and wrist classes [158]. Alternatively, regression control's main advantage is that it offers the potential for fluid

prosthetic wrist/hand movements, even during transitions [31]. To achieve such fluidity, however, users must execute complex training routines that require dynamic elicitation of EMG signals with varying intensities. Choosing classification- over regression-based control evidently comes with a trade-off: movement reliability and model simplicity, over movement fluidity.

Control reliability remains a research goal, particularly as a problem known as the *limb position effect* critically impedes device reliability [15], [16], [17], [18]. Here, surface EMG signals change when a prosthesis or other wearable device is used in untrained limb positions. It has been established that when the physical conditions for pre-training and training are dissimilar, muscle coactivation patterns can be introduced during limb movements and incorrect decoding of a user's intent can result [18]. Muscle coactivations are introduced even among those without amputation, normally evidenced in high limb positions [10], [18]. The limb position reliability implications to prosthesis control are well acknowledged, yet remain largely unsolved [14]. To definitively solve it, a control model would have to be trained in every conceivable limb position, but in doing so would require an excessively long training time. Instead, training routines performed in selected low, midway, and high positions have been introduced to combat the limb position effect problem, with their models yielding statistically significant control improvements [15].

Researchers have also begun to add positional sensors (worn on users' residual limbs during model training and testing) to address the problem, and have found that inertial measurement units (IMUs) can capture pertinent limb position data [15], [16], [21], [22], [23]. Our earlier work built upon this finding and successfully combined EMG and IMU data to train deep learning pattern recognition-based controllers [10]. Specifically, we developed a new type of recurrent convolutional neural network (RCNN), intended for upper limb prosthesis control. RCNNs are a network architecture for deep learning, capable of learning directly from data and handling large amounts of multimodal data [27], [28]. This makes them well-suited for training with EMG and IMU data from multiple limb positions, without requiring researchers to supply engineered features—instead, features are learned [43], [70], [71]. RCNNs have also proven to be advantageous as they can handle the intrinsic time-varying nature of muscle signals [72]. Our work capitalized on these benefits and established that position-aware RCNN controllers show promise towards mitigating the limb position effect [10]; as corroborated by our research that focused on the movements of participants without limb difference [10], [11], [90], [99].

Our body of control research led us to consider whether a more advanced *RCNN-based classification* and/or *RCNN-based regression* controller could offer improved control and functionality over earlier pattern recognition-based approaches that focused on reliability and ease-of-implementation at the expense of movement fluidity. We aimed to recommend an RCNN-based controller that uses EMG and IMU sources of movement/positional data to effectively mitigate the limb position effect. Solving this prevalent EMG-based device control problem would make inroads towards acceptance of future control solutions—by rehabilitation clinicians and users alike.

In the work presented herein, we investigate whether an RCNN classification controller (RCNN-Class) and/or an RCNN regression controller (RCNN-Reg) might offer enhanced myoelectric prosthesis control versus a conventional linear discriminant analysis (LDA) classification counterpart. The latter is commonly used to control upper limb prostheses, and as such, is often adopted in research as a baseline for comparison to other controllers [6], [10]. This work equipped non-disabled participants with a simulated device, which has been shown to be a reasonable proxy for actual myoelectric prosthesis use [77]. With the device donned, they trained and tested either RCNN-Class or RCNN-Reg, along with an LDA baseline controller (LDA-Baseline). Participants executed functional tasks across multiple limb positions during testing, with EMG and IMU data collected. Task performance metrics analysis relied on motion capture data, control characteristics analysis was based on the prosthesis' motor data, and participants' control experiences were gauged from their responses to surveys. Both RCNN-Class and RCNN-Reg showed improved functional task performance over LDA-Baseline. RCNN-Reg, however, offers two fundamental control advantages to users: (1) it mitigates the limb position effect, plus (2) it reliably provides smooth and simultaneous device movements. Our RCNN regression-based findings contribute to the body of myoelectric prosthesis control research by presenting a novel and advanced control approach that provides fluid movements that more closely approximate those of an intact wrist and hand. What follows are details about this work's experimental methods, results, and promising regression-based control outcomes.

## 7.2. Methods

### 7.2.1. Overview

In this work, we compared two RCNN-based pattern recognition controllers to an LDA classification controller baseline, which applies probability theory to discover patterns in EMG data and uses engineered features to inform control [6], [25]. We considered the control performance offered by: (1) RCNN-Class versus LDA-Baseline, and (2) RCNN-Reg versus LDA-Baseline. For our investigation, three distinct controller testing sessions were undertaken, as described below.

**A) An RCNN-Class Session** (outlined in Figure 7-2A) required participants to don a gesture control armband equipped with both EMG and IMU sensors, plus a simulated prosthesis. While wearing this equipment, participants performed a training routine that involved *static*, isotonic forearm muscle contractions in *four* limb positions (in doing so, they trained RCNN-Class's model). After learning how to control the device through practice, participants performed functional tasks with motion capture data recorded.

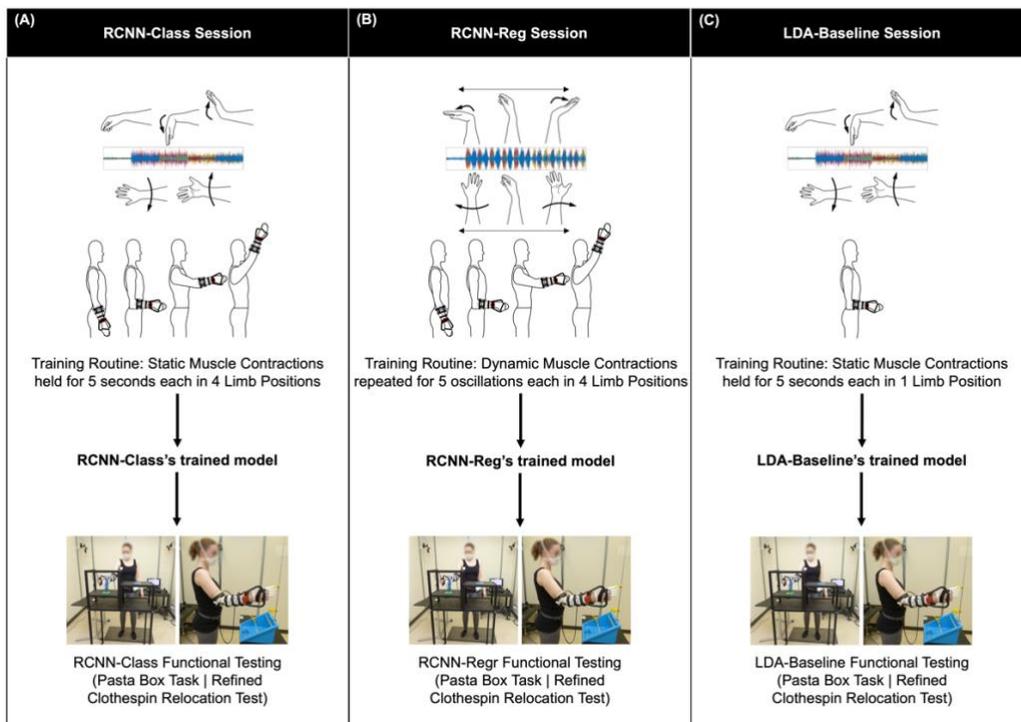


Figure 7-2. Overview of controller testing sessions: (A) RCNN-Class Session, (B) RCNN-Reg Session, and (C) LDA-Baseline Session. Note that for all training routine and testing execution, participants wore a gesture control armband (equipped with EMG and IMU sensors), along with a simulated prosthesis.

**B) An RCNN-Reg Session** (outlined in Figure 7-2B) required participants to don the gesture control armband and simulated prosthesis, after which they performed a training routine that involved *dynamic*, isotonic forearm muscle contractions in *four* limb positions (in doing so, they trained RCNN-Reg's model). After learning how to control the device through practice, participants performed functional tasks with motion capture data recorded.

**C) An LDA-Baseline Session** (outlined in Figure 7-2C) also required participants to don the gesture control armband and simulated prosthesis, after which they performed a training routine that involved *static*, isotonic forearm muscle contractions in only *one* limb position (in doing so, they trained LDA-Baseline's model). Notably, this training routine did not involve muscle contractions in multiple limb positions as required by the RCNN-based controllers' models, for three reasons: (1) training in only one limb position mimics standard prosthesis controller training [18]; (2) the goal of this research was not to investigate a position-aware LDA-based controller, but rather to investigate the potential effectiveness of RCNN-based controllers for mitigating the limb position effect (as they can combine large volumes of EMG and IMU data); and (3) we have previously found position-aware RCNN-based classification to better mitigate the limb position effect versus position-aware LDA-based classification [10]. After learning how to control the device through practice, participants performed the same functional tasks as used with the RCNN-based controller testing, with motion capture data recorded.

### 7.2.2. Participants

A total of 16 participants were recruited for this study. All participants provided written informed consent, as approved by the University of Alberta Health Research Ethics Board (Pro00086557). Each participant completed two testing sessions, with an RCNN-based controller tested in one session and LDA-Baseline tested in the other. A washout period of at least seven days was included between the two sessions to ensure that participants forgot details of their first session's controller. The 16 participants were split into two groups.

The first group compared **RCNN-Class versus LDA-Baseline**. Eight participants were recruited for this group. They had a median age of 25 (range: 22–29) and median height of 173 cm (range: 167–181 cm), four were male, four were female, and all were right-handed. One participant had minimal previous experience with EMG pattern recognition control. Four participants in this group

trained and tested RCNN-Class in their first session, whereas the other four participants trained and tested LDA-Baseline in their first session. Participants trained and tested the remaining controller in their second session, with a median of 27 days between the first and second sessions (range: 13–42 days).

The second group compared **RCNN-Reg versus LDA-Baseline**. Eight participants were recruited for this group. They had a median age of 24 (range: 19–27) and median height of 176 cm (161–193 cm), five were male, three were female, seven were right-handed and one was ambidextrous. No participants had previous experience with EMG pattern recognition control. Four participants in this group trained and tested RCNN-Reg in their first session, whereas the other four participants trained and tested LDA-Baseline in their first session. Participants then trained and tested the remaining controller in their second session, with a median of 17.5 days between the first and second sessions (range: 7–25 days).

### 7.2.3. Muscle Signal Data Collection

#### 7.2.3.1. Myo Gesture Control Armband

All participants wore a Myo gesture control armband (Thalmic Labs, Kitchener, Canada—discontinued) over their largest forearm muscle bulk [15], to facilitate capture of EMG signal data during controller training and testing. The armband was worn at approximately the upper third of participants’ right forearm, as shown in Figure 7-3A (with the top of the armband at a median of 28% of the way down the forearm from the medial epicondyle to the ulnar styloid process). The Myo armband contained eight surface electrodes to collect EMG data at a sampling rate of 200

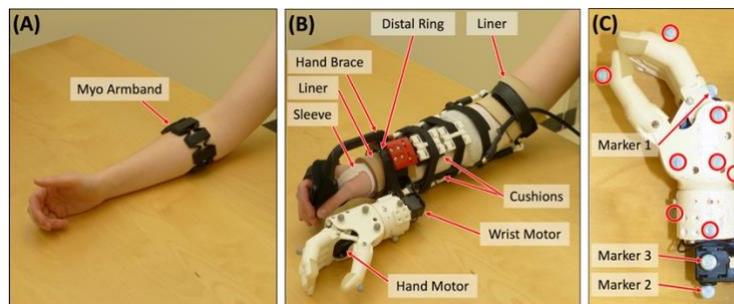


Figure 7-3. (A) Myo armband on a participant’s forearm; (B) simulated prosthesis on a participant’s forearm, with labels indicating the sleeve, two pieces of liner, hand brace, distal ring, cushions, wrist motor, and hand motor; and (C) motion capture markers affixed to the simulated prosthesis, with the eight motion capture markers that remained attached to the hand circled, and the three additional individual markers for the ski pose calibration are labelled. Adapted from Williams et al. [90].

Hz. It also contained one IMU to collect limb position data (three accelerometer, three gyroscope, and four quaternion data streams) at 50 Hz. Note that from the IMU, only the accelerometer data streams were used to ascertain limb position. Myo Connect software was used to stream and record EMG and IMU data in Matlab.

#### 7.2.3.2. Simulated Prosthesis

The simulated prosthesis used in this study was the 3D-printed Modular-Adaptable Prosthetic Platform (MAPP) [92] (shown in Figure 7-3B). It was fitted to each participant's right arm, to simulate transradial prosthesis use. The MAPP's previously published design [92] was altered to improve wearer comfort in our study—the distal ring was made to resemble the oval shape of a wrist and the hand brace was elongated so that the distal ring would sit more proximally on the wearer's wrist. A non-proprietary 3D-printed robotic hand [93] was affixed to the MAPP beneath the participant's hand. Wrist rotation capabilities were also added to the device. Hand and wrist movements (that is, with two degrees of freedom) were powered by two Dynamixel MX Series motors (Robotis Inc., Seoul, South Korea).

After placement of the Myo gesture control armband, participants donned a thin, protective sleeve. The MAPP was donned over the sleeve and affixed with Velcro straps. Gel-coated pieces of fabric liner (that is, with thermoplastic elastomer) were placed inside the MAPP's distal ring and just above the participant's elbow to ensure a comfortable fit. In addition, 3D-printed cushions, made of Ninjaflex Cheetah filament (Ninjatek, Inc.), were placed throughout the device socket (shown in Figure 7-3B). The MAPP was checked for secureness, a visual inspection was performed to ensure that no components were loose, and a final participant comfort check was conducted verbally.

#### 7.2.3.3. EMG and Accelerometer Data Processing

The EMG data from the Myo armband were filtered using a high pass filter with a cutoff frequency of 20 Hz (to remove movement artifacts), as well as a notch filter at 60 Hz (to remove electrical noise). The accelerometer data streams were upsampled to 200 Hz (using previous neighbour interpolation) to align them with the corresponding EMG data. Data were then segmented into windows (160-millisecond with a 40-millisecond offset).

## 7.2.4. Controller Descriptions: Model Architectures, Training Routines, & Implementation

Each of the RCNN-based controllers and their LDA-Baseline counterpart are described below. The two RCNN-based controller descriptions include details about its model's architecture, training routine, hyperparameters used, and number of weights. LDA-Baseline's description includes details about its statistical model implementation, training routine, features used, and number of coefficients. For each of the RCNN-based controllers presented, all hyperparameters were determined using Bayesian optimization—this includes the number of convolution layers, number of filters, filter size, pooling size, and early stopping criteria of the model. Each model's optimization process was performed in two steps. First, an initial broad range of values were used to optimize each hyperparameter. Thereafter, values were refined using a narrower range, centered at earlier optimized values. Given that each model's hyperparameters were optimized separately, their resulting architectures had a unique number of such with different training durations to achieve their best possible accuracies—that is, the models did not require the same number of weights nor training times to each achieve this.

### 7.2.4.1. RCNN-Class Controller

**Model Architecture**—RCNN-Class's model architecture consisted of 23 layers, as illustrated in Figure 7-4A. In this model, a sequence input layer first received and normalized the training data. Then, a sequence folding layer was used, allowing convolution operations to be performed independently on each window of EMG and accelerometer data. This was followed by a block of four layers: a 2D convolution, a batch normalization, a rectified linear unit (ReLU), and an average pooling layer. This block of layers was repeated twice more. Each of the three average pooling layers had a pooling size of 1x4. A block of three layers followed: a 2D convolution, a batch normalization, and a ReLU layer. The optimal number of filters in the convolution layers were determined to be 16, 32, 64, and 8, respectively, and each had a filter window size of 1x4. The next layers included a sequence unfolding layer (to restore the sequence structure), a flatten layer, a long short-term memory (LSTM) layer, and a fully connected layer. Finally, a softmax layer and classification layer yielded the final class predictions. RCNN-Class's model had a total of 76,482 weights. Note that to prevent model overfitting, an early stopping patience hyperparameter (criteria) was set. In doing so, model training would automatically stop when the validation loss

(calculated every 50 iterations) increased four times. This overfitting mitigation method chosen for this control model, is similar to that used in other works [71].

**Model Training Routine**—Participants followed onscreen instructions, performing static muscle contractions in five wrist positions (rest, flexion, extension, pronation, and supination; shown in Figure 7-2A), for five seconds each. The muscle contractions were performed twice in four limb positions: arm at side, elbow bent at 90°, arm straight out in front at 90°, and arm up 45° from vertical (shown in Figure 7-2A). This multi-limb-position training routine was similar to those used in other real-time control studies aiming to mitigate the limb position effect [10], [15], [23]). It took approximately 200 seconds. The resulting EMG and accelerometer data, plus corresponding classes of muscle contractions, were used to train RCNN-Class’s model. A median total of 4,836 unique samples (where samples are defined as windows of 8 EMG and 3 accelerometer channels by 32 time stamps) were used for model training. Model training took a median of 5,275 iterations (meaning that a total of 25.5 million samples were observed), which required a median of 104.8 seconds to complete (with computer specifications detailed in the Control Model Implementation section).

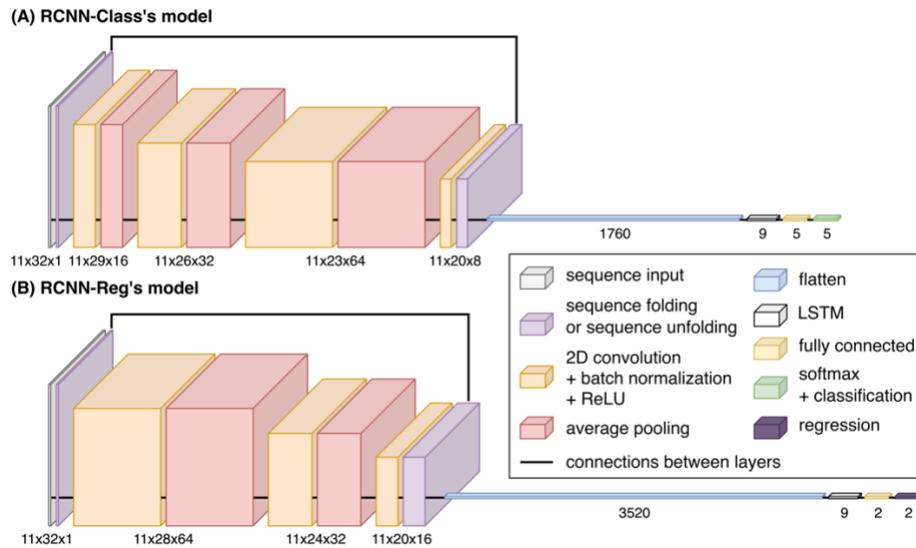


Figure 7-4. Architecture of (A) RCNN-Class’s model and (B) RCNN-Reg’s model, including the size of each layer and connections between layers.

### 7.2.5. RCNN-Reg Controller

**Model Architecture**—RCNN-Reg’s model architecture consisted of 18 layers, as illustrated in Figure 7-4B. The first layers of this model are a sequence input layer and a sequence folding layer. These were followed by a block of four layers: a 2D convolution, a batch normalization, a ReLU, and an average pooling layer. This block of layers was repeated once more. Each of the two average pooling layers had a pooling size of 1x4. A block of three layers followed: a 2D convolution, a batch normalization, and a ReLU layer. The optimal number of filters in the convolution layers were determined to be 64, 32, and 16, respectively, and each had a filter window size of 1x5. The next layers included a sequence unfolding layer (to restore the sequence structure), a flatten layer, an LSTM layer, and a fully connected layer. Finally, a regression layer yielded the final predictions. RCNN-Reg’s model had a total of 140,556 weights. Note that to prevent model overfitting, an early stopping patience hyperparameter (criteria) was set. In doing so, model training would automatically stop when the validation loss (calculated every 50 iterations) increased once.

**Model Training Routine**—Participants followed onscreen instructions, performing three types of muscle contractions (shown in Figure 7-2B):

1. The wrist was held at rest for five seconds;
2. Dynamic muscle contractions, oscillating five times between full wrist flexion (that is, with a strong contraction that did not introduce discomfort) and corresponding full wrist extension; and
3. Dynamic muscle contractions, oscillating five times between full forearm pronation and full forearm supination.

All such muscle contractions were performed twice in four limb positions: arm at side, elbow bent at 90°, arm straight out in front at 90°, and arm up 45° from vertical (shown in Figure 7-2B). This multi-limb-position routine was developed in our previous offline research [10] and took approximately 300 seconds. Note that although this training routine was longer than that used for RCNN-Class, it is comparable as it allowed the model to be exposed to intermediate points between full muscle contractions. Rather than employing classes of muscle contractions in this work, two arrays with values between -1 and 1 were used to represent the flexion-extension and

the pronation-supination degree of freedoms, respectively. For the static rest muscle contractions, both arrays contained zeros. For the flexion-extension oscillations, the first array contained sinusoidal values between -1 (representing full flexion) and 1 (representing full extension), and the second array contained zeros. For pronation-supination oscillations, the first array contained zeros and the second array contained sinusoidal values between -1 (representing full pronation) and 1 (representing full supination). The resulting EMG and accelerometer data, plus corresponding values, were used to train RCNN-Reg's model. A median total of 7,420 unique samples were used for model training. Model training took a median of 500 iterations (meaning that a total of 3.7 million samples were observed), which required a median of 21.4 seconds to complete.

#### 7.2.5.1. LDA-Baseline Controller

**Model Details**—Four commonly used EMG features were chosen for implementation of this baseline classifier's model: mean absolute value, waveform length, Willison amplitude, and zero crossings [9]. These features were calculated for each channel within each window of EMG data. A pseudo-linear LDA discriminant type was used, given that columns of zeros were occasionally present in some classes for some features (including Willison amplitude and zero crossings). LDA-Baseline's model had a total of 330 coefficients.

**Model Training Routine**—Participants followed onscreen instructions, performing muscle contractions in five wrist positions (shown in Figure 7-2C), for five seconds each. The muscle contractions were performed twice, with the participants' elbow bent at 90° (shown in Figure 7-2C). This single-position routine mimicked standard myoelectric prosthesis training [18] and took approximately 50 seconds. The resulting EMG data and corresponding classes of muscle contractions were used to train LDA-Baseline's model. Features calculated from a median total of 1,210 unique samples (where samples are defined as windows of 8 EMG channels by 32 time stamps) were used for model training. Model training required a median of 0.7 seconds to complete.

#### 7.2.5.2. Control Model Implementation

Each of the three control models implemented in this study was trained using Matlab software running on a computer with an Intel Core i9-10900K processor (3.70 GHz), a

NVIDIA GeForce RTX 2080 SUPER graphics card with 8GB GDDR6, and 128 GB of RAM. RCNN-Class's, RCNN-Reg's, and LDA-Baseline's models were trained in median times of 104.8, 21.4, and 0.7 seconds, respectively. After completion of all model training, the simulated prosthesis was programmatically controlled as follows:

1. Matlab code was written to receive EMG and accelerometer data, from which the controller predicted intended wrist movements. Note that for RCNN-Reg, additional code was written to smooth predictions using a moving average filter (averaging the current prediction with the previous prediction) [10], [79].
2. Matlab code was also written to enable predicted motor instruction transmission to the device wrist/hand, based on resulting classifications—that is, via brachI/Oplexus software [94], where received flexion signal data were translated to hand close, extension to hand open, pronation to move wrist in a counter-clockwise rotation, and supination to move wrist in a clockwise rotation. Note that for RCNN-Reg, small prediction values were suppressed to 0 as required [10], [42], with a median value of  $\pm 0.05$  (range: 0.02–0.1), out of a total signal range of -1 to 1.
3. brachI/Oplexus relayed the corresponding instructions to the simulated prosthesis' motors. The motor instructions and positions were recorded with a sampling rate of 50 Hz.

#### 7.2.6. Simulated Device Control Practice & Testing Eligibility

Recall that for each of the RCNN-Class, RCNN-Reg, and LDA-Baseline testing sessions, participants wore a gesture control armband (equipped with EMG and IMU sensors), along with a simulated prosthesis. With all such equipment donned and during each testing session, participants took part in 40-minute control practice periods. During this period, they were taught how to operate the simulated prosthesis using isometric muscle contractions, under three conditions:

1. Controlling the hand open/close while the wrist rotation function was disabled. They practiced grasping, transporting, and releasing objects at varying heights.
2. Controlling wrist rotation while the hand open/close function was disabled. They practiced rotating objects at varying heights.

3. Controlling the hand open/close function in concert with the wrist rotation function. They practiced tasks that involved grasping, transporting, rotating, and releasing objects at varying heights.

For RCNN-Reg control practice, each of the three conditions also required participants to practice controlling device movement velocity (that is, trying to slow down and speed up device movements). Furthermore, Condition 3 included an opportunity to practice simultaneous control of the two degrees of freedom (that is, performing wrist rotation and hand open/close at the same time).

Following all practice periods, participants were tested to determine whether they could reliably control the simulated prosthesis. For such testing, two cups were situated in front of them at two different heights, with a rubber ball in one of the cups. Participants were asked to pour the ball between the two cups, and instances when they dropped the ball or a cup were recorded. If participants could not complete at least 10 pours with a success rate of at least 75% within 10 minutes, the session was ended, and they were removed from the study. All participants passed this test.

### 7.2.7. Motion Capture Setup & Kinematic Calibrations

After participants were deemed eligible for controller testing, the following motion capture steps were undertaken.

**Step 1: Motion Capture Setup**—A 10-camera OptiTrack Flex 13 motion capture system (Natural Point, OR, USA) was used to capture participant movements and task objects at a sampling rate of 120 Hz. Eight individual markers were placed on the simulated prosthesis hand, circled in Figure 7-3C (one on the thumb, one on the index finger, and the remaining six throughout the back and side of the hand to ensure reliable rigid body tracking). Rigid marker plates were also placed on each participant's right forearm (affixed to the simulated prosthesis socket), upper arm, and thorax, in accordance with Boser et al.'s cluster-based marker model [150].

**Step 2: Kinematic Calibrations**—Each participant was required to perform two kinematics calibrations. As per Boser et al., the first calibration called for participants to hold an anatomical pose [150], for capture of the relative positions of the hand markers and motion capture marker

plates when wrist rotation and shoulder flexion/extension angles were at 0°. The second calibration required participants to hold a ski pose [150], for the purpose of refining wrist rotation angles. Here, three additional individual markers were affixed to the simulated prosthesis, as shown in Figure 7-3C:

1. One marker placed on the top of the prosthesis' hand motor, with the device hand closed
2. One marker placed on the bottom of the prosthesis' wrist motor, forming a line with the first marker (to represent the axis about which the wrist rotation occurred)
3. One marker placed on the side of the prosthesis' wrist motor (to create a second axis, perpendicular to the axis of wrist rotation)

Upon completion of the two kinematics calibrations, all Step 2 markers were removed. What remained were only those markers affixed during Step 1 for data collection purposes.

## 7.2.8. Controller Testing

Testing required participants to execute functional tasks that mimicked activities of daily living across multiple limb positions, with motion capture data and participant survey results as outputs.

### 7.2.8.1. Functional Task Execution

Motion capture data were collected while participants executed the following functional tasks.

**Pasta Box Task (Pasta)**—Participants were required to perform three distinct movements, where they transported a pasta box between a 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> location (a side table and two shelves at varying heights on a cart, including across their midline) [95]. The task setup is shown in Figure 7-5A. Motion capture markers were placed on all task objects, as per Valevicius et al. [95]. Participants performed a total of 10 Pasta trials. If participants dropped the pasta box, placed it incorrectly, performed an incorrect movement sequence, or hit the frame of the task cart, the trial was not analyzed. Pasta was the first of two functional tasks performed as it was considered easier.

**RCRT**—Participants were required to perform three distinct movements using clothespins. They moved three clothespins between 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> locations on horizontal and vertical bars [96]. To simplify trial execution, RCRT was split into RCRT Up and RCRT Down trials. The task setup for these trails is shown in Figure 7-5B. During Up trials, participants moved the three clothespins from the horizontal bar to the vertical bar, and during Down trials, they moved the clothespins

from the vertical bar to the horizontal bar. A height adjustable cart was set up—with its top surface height situated at 65 cm below the participant’s right shoulder. This setup ensured that the top of each participants’ shoulder was aligned with the midpoint between the top two targets on the vertical bar used in the RCRT Up and Down trials. Motion capture markers were placed on all task objects, as per our earlier research [90]. Participants performed a total of 10 Up trials and 10 Down trials. If participants dropped a clothespin, placed it incorrectly, or performed an incorrect movement sequence, the trial was not analyzed. Performance of RCRT Up and Down trials were alternated, beginning with RCRT Up.

### 7.2.8.2. Perceived Control via Participant Survey Administration

At the end of every controller testing session, participants completed two surveys: the NASA-TLX [145] and a usability survey [147]. The former was administered using the official **NASA-TLX** tablet application, where participants scored their device control workload demand on a continuous rating scale with endpoint anchors of low and high. The **usability survey** was administered on paper, where participants marked their usability scores on a continuous rating scale with endpoint anchors of 0 and 5. In their second session, participants were *not* reminded of their survey responses from their first session.

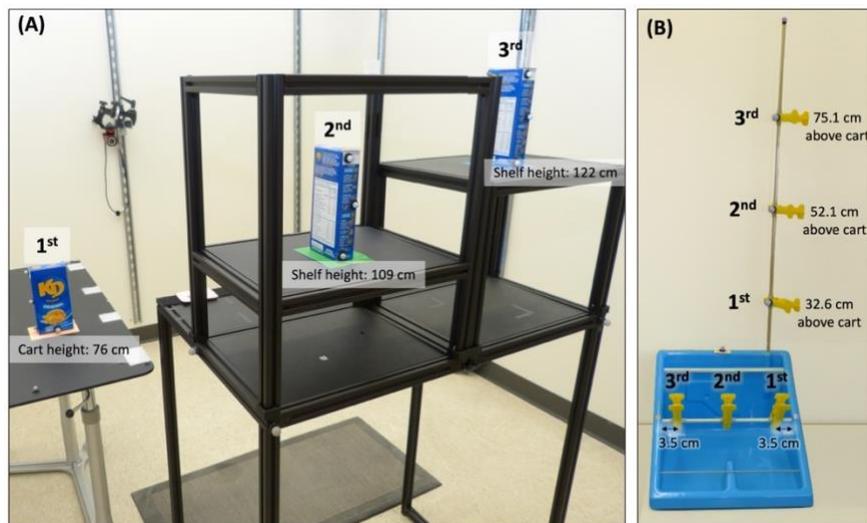


Figure 7-5. Task setup for (A) Pasta and (B) RCRT Up and Down, including task dimensions. In panel (A), the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> pasta box locations are labelled. The pasta box movement sequence is 1<sup>st</sup> → 2<sup>nd</sup> → 3<sup>rd</sup> → 1<sup>st</sup> locations. In panel (B), the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> clothespin locations on the horizontal and vertical bars are labelled. The clothespin movement sequences in RCRT Up are horizontal 1<sup>st</sup> → vertical 1<sup>st</sup>, horizontal 2<sup>nd</sup> → vertical 2<sup>nd</sup>, and horizontal 3<sup>rd</sup> → vertical 3<sup>rd</sup> locations. The clothespin movement sequence in RCRT Down follows the same order, but with each clothespin moved from vertical to horizontal locations. Adapted from Williams et al. [90].

## 7.2.9. Analysis of Controller Testing Data

Statistical data analysis was undertaken after the following steps were completed: processing of the motion capture data (with some measures estimated); finalization of estimated measures; task segmentation; and calculation of control assessment metrics.

### 7.2.9.1. Motion Capture Data Processing

Motion capture data were cleaned and filtered. As per Valevicius et al. [95], grip aperture was estimated using the distance between the motion capture markers on the simulated prosthesis' index finger and thumb. In addition, a 3D object representing the simulated prosthesis' hand was generated using the remaining 6 hand markers, for the purpose of calculating upcoming metrics [95]. Next, using calculations modified from Boser et al. [150], wrist rotation angles were estimated using the forearm and hand markers, whereas shoulder flexion/extension angles were calculated using the upper arm and thorax motion markers.

### 7.2.9.2. Finalization of Grip Aperture and Wrist Rotation Measures

The grip aperture and wrist rotation measures were finalized using data from the simulated prosthesis' two motors. As small (yet informative) adjustments in the positions of these motors may not have been detected by the motion capture cameras, we chose not to ignore them. First, the positions of the motors were upsampled to 120 Hz using linear interpolation. Then, the estimated measures were finalized—*Grip aperture*: the motion-capture-estimated grip apertures were fitted to a trinomial curve, to facilitate transforming the hand motor data to final grip aperture measures. *Wrist rotation*: the motion-capture-estimated wrist rotation angles were fitted to a binomial curve, to facilitate transforming the wrist motor data to final wrist rotation angle measures.

### 7.2.9.3. Task Segmentation

The task data were segmented in accordance with Valevicius et al. [95]. For each *task*, the data from each *trial* were first divided into distinct *movements 1, 2, and 3* based on hand velocity and the velocity of the pasta box/clothespins during transport.

**Pasta** Movements 1, 2, and 3 differentiated between: (1) reaching for the pasta box at its 1<sup>st</sup> location, grasping it, transporting it to its 2<sup>nd</sup> location, releasing it, and moving their hand back to a home position; (2) reaching for the pasta box at the 2<sup>nd</sup> location, grasping it, transporting it to its

3<sup>rd</sup> location, releasing it, and moving their hand back to the home position; and (3) reaching for the pasta box at the 3<sup>rd</sup> location, grasping it, transporting it back to the 1<sup>st</sup> location, releasing it, and moving their hand back to the home position.

**RCRT Up** Movements 1, 2, and 3 differentiated between: (1) reaching for the 1<sup>st</sup> clothespin at its 1<sup>st</sup> horizontal location, grasping it, transporting it to its 1<sup>st</sup> vertical location, releasing it, and moving their hand back to a home position; (2) reaching for the 2<sup>nd</sup> clothespin at its 2<sup>nd</sup> horizontal location, grasping it, transporting it to its 2<sup>nd</sup> vertical location, releasing it, and moving their hand back to the home position; and (3) reaching for the 3<sup>rd</sup> clothespin at its 3<sup>rd</sup> horizontal location, grasping it, transporting it to its 3<sup>rd</sup> vertical location, releasing it, and moving their hand back to the home position.

**RCRT Down** Movements 1, 2, and 3 differentiated between: (1) reaching for the 1<sup>st</sup> clothespin at its 1<sup>st</sup> vertical location, grasping it, transporting it to its 1<sup>st</sup> horizontal location, releasing it, and moving their hand back to a home position; (2) reaching for the 2<sup>nd</sup> clothespin at its 2<sup>nd</sup> vertical location, grasping it, transporting it to its 2<sup>nd</sup> horizontal location, releasing it, and moving their hand back to the home position; and (3) reaching for the 3<sup>rd</sup> clothespin at its 3<sup>rd</sup> vertical location, grasping it, transporting it to its 3<sup>rd</sup> horizontal location, releasing it, and moving their hand back to the home position.

Then, the data from each of the three movements were further segmented into *five phases* of Reach, Grasp, Transport, Release, and Home (note that the Home phase was not used for data analysis). Finally, *two movement segments* of Reach-Grasp and Transport-Release were defined for select metrics analysis. Six final levels for data analysis resulted: *controller* (either RCNN-Class, RCNN-Reg, or LDA-Baseline), *task* (either Pasta, RCRT Up, or RCRT Down), *trial* (1–10), *movement* (1–3), *movement segment* (Reach-Grasp or Transport-Release), and *phase* (Reach, Grasp, Transport, or Release).

#### 7.2.9.4. Control Assessment Metrics Calculated

The *Suite of Myoelectric Control Evaluation Metrics*, introduced in our previous work [99], was used to compare prosthesis control resulting from RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline. A summary of these metrics can be found in Table 7-1.

Table 7-1. Summary of metrics used in this study, including details of whether a smaller or larger value indicates a control improvement.

Metric Category	Metric	Metric Description	Control Improvement Indicator	
Task Performance	Success Rate (%)	Percent of trials that are error-free	Larger	
	Trial Duration (sec)	Elapsed time for each trial	Smaller	
	Phase Duration (sec)	Elapsed time for each phase	Smaller	
	Relative Phase Duration (%)	Elapsed time for each phase, relative to the elapsed time for a Reach-Grasp-Transport-Release movement	Grasp, Release: smaller Reach, Transport: larger	
	Peak Hand Velocity (mm/s)	Maximum velocity of the hand while moving	Larger	
	Hand Distance Travelled (mm)	Total distance travelled by the hand while moving	Smaller	
	Hand Trajectory Variability (mm)	How much the hand movement path varies between trials	Smaller	
Control Characteristics	Total Grip Aperture Movement (mm)	Total amount of grip aperture variation	Smaller	
	Total Wrist Rotation Movement (deg)	Total amount of wrist rotation angle variation	Smaller	
	Number of Grip Aperture Adjustments	Number of times that grip aperture variation commences or changes direction	Smaller	
	Number of Wrist Rotation Adjustments	Number of times that wrist rotation angle variation commences or changes direction	Smaller	
	Grip Aperture Plateau (sec)	Amount of time during which the grip aperture remains open before closing to grasp a task object	Smaller	
	Simultaneous Wrist-Shoulder Movements (%)	Percent of the phase during which the wrist rotation is controlled while the shoulder is moving	Larger	
User Experience	Total Muscle Activity		Smaller	
	NASA-TLX	Mental Demand	Workload demand resulting from each controller	Smaller
		Physical Demand		Smaller
		Temporal Demand		Smaller
		Performance		Smaller
		Effort		Smaller
		Frustration		Smaller
	Usability Survey	Intuitiveness	Usability of each controller	Larger
		Effectiveness – Pasta		Larger
		Effectiveness – RCRT		Larger
Reliability		Larger		

To identify occurrences where the limb position effect impeded control, we analyzed our resulting Control Characteristics metrics from the suite, as outlined in our previous work [99]—the thresholds used for identification of the limb position effect remained the same, with the total muscle activity metric eliminated from analysis thereof. Specifically, this present work’s limb position effect analysis considered trends across the Control Characteristics metrics’ medians and interquartile ranges (IQRs) for each task’s three movements, with larger medians and/or larger interquartile ranges providing evidence of degraded control. Figure 7-6A illustrates an example where the limb position effect was identified by the increasing medians and IQRs from movement 1 (in the lowest limb position) to movement 3 (in the highest limb position). Conversely, Figure 7-6B illustrates an example where the limb position effect was *not* identified.

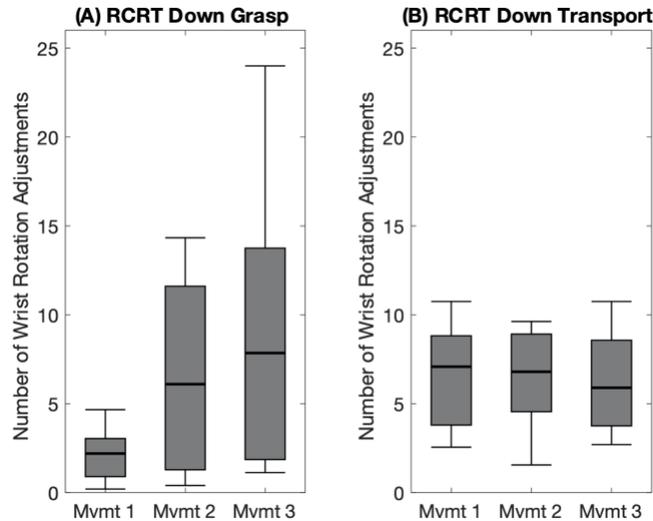


Figure 7-6. Box plots indicating LDA-Baseline number of wrist rotation adjustments in (A) RCRT Down Grasp and (B) RCRT Down Transport of each task movement (Mvmt). Medians are indicated with thick lines, and interquartile ranges are indicated with boxes.

Finally, to assess whether participants took advantage of RCNN-Reg’s simultaneous and proportional velocity control capabilities, we calculated two additional metrics from recorded motor instructions. To assess whether participants capitalized on RCNN-Reg’s **simultaneous control** capabilities, simultaneous wrist-grip movements were calculated—the percent of each phase in which the wrist rotation and the grip aperture velocities were simultaneously moving. Of note, this metric was only calculated for RCRT Up and RCRT Down tasks (given that wrist rotation movements were not necessary for Pasta). Furthermore, the metric was only applicable to Reach, Grasp, and Release phases (given that grip aperture adjustments were not necessary in Transports). To assess whether participants capitalized on RCNN-Reg’s **proportional velocity control** capabilities, instances where velocity fluctuated between 0 and full speed were counted—as indicated by the percentage of each task in which wrist rotation and grip aperture respectively varied. These two motor-recorded metrics were only calculated for RCNN-Reg, given that neither simultaneous control nor proportional velocity control were possible under RCNN-Class or LDA-Baseline alternatives.

#### 7.2.9.5. Statistical Analysis

To compare control resulting from RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline, the following statistical analyses were performed using all Task Performance, Control Characteristics, and User Experience metrics:

**For metrics that were analyzed at the phase or movement segment level**—Participants' results were first averaged across trials and movements. If results were normally-distributed, a two-factor repeated-measures analysis of variance (RMANOVA) was conducted using the factors of controller and phase/movement segment. When the resulting controller effects or controller-phase/movement segment interactions were deemed significant (that is, when the Greenhouse-Geisser corrected p value was less than 0.05), pairwise comparisons between the controllers were conducted. If results were not normally-distributed, the Friedman test was conducted. When the resulting p value was less than 0.05, pairwise comparisons between the controllers were conducted. Pairwise comparisons (t-test/Wilcoxon sign rank test) were deemed significant if the p value was less than 0.05.

**For metrics that were analyzed at the trial level**—Participants' results were first averaged across trials, then pairwise comparisons were conducted as detailed above.

**For metrics that were analyzed at the task or controller level**—Pairwise comparisons were conducted as detailed above.

One participant's data were not included in this study's statistical analysis. That participant experienced difficulties with LDA-Baseline control in their second testing session (comparing RCNN-Reg versus LDA-Baseline). Although they passed their practice test, they only completed three error-free trials of Pasta and were unable to complete any error-free trials of RCRT Up or Down. Consequently, their data stemming from all controller testing sessions were excluded from analysis.

### 7.3. Results

The Suite of Myoelectric Control Evaluation Metrics [99] was used in this comparative controller study. The suite includes three broad categories of metrics: task performance, control characteristics, and user experience. Through an analysis of these metrics, numerous control

findings were uncovered. These findings included identification of significant improvements in RCNN-based controller performance over LDA-Baseline, metrics-substantiated evidence of the limb position effect, indications that prosthesis control improvements in high limb positions can be offered by RCNN-based controllers, and that RCNN-Reg’s simultaneous and proportional velocity control capabilities yield overall improved device performance.

### 7.3.1. Significant Differences Between RCNN-Based Controllers and LDA-Baseline

Instances of significant improvements for RCNN-based control over LDA-Baseline are summarized in Figure 7-7—48 Task Performance and 67 Control Characteristics metrics for Pasta, RCRT Up, and RCRT Down tasks are presented. Here, 11 out of 115 metrics showed significant controller performance improvement by RCNN-Class over LDA-Baseline, and 38 out of 115 metrics showed significant controller performance improvements by RCNN-Reg versus LDA-Baseline. No significant differences between controllers were identified in User Experience metrics. All controller comparison results can be found in Appendix D.

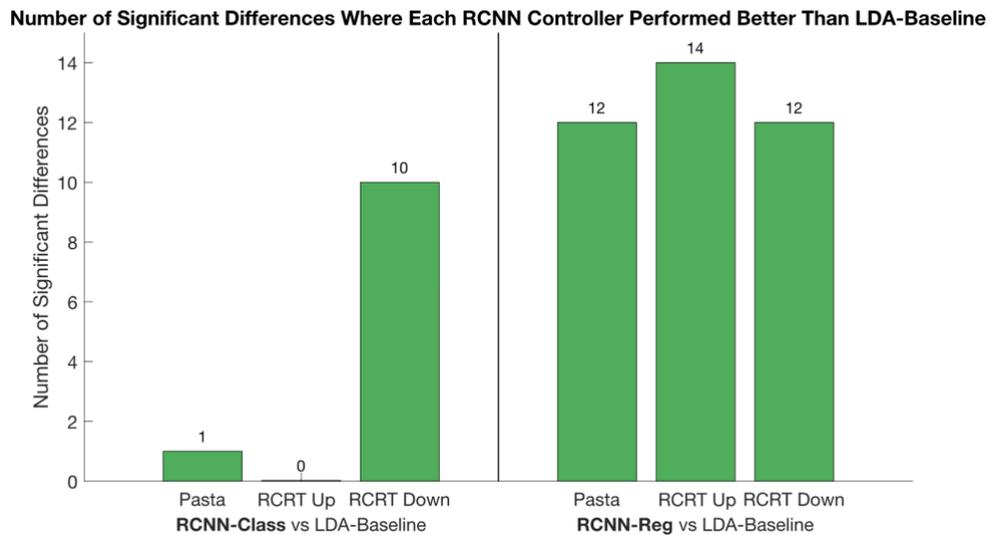


Figure 7-7. Number of significant differences between RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline in Task Performance and Control Characteristics metrics. An average tally of performance improvement results is presented per task.

Two examples are presented below to illustrate controller outcomes during movements in high limb positions. One example presents a discrete metric, the other presents a continuous metric. Note that the LDA-Baseline results presented in these examples vary slightly between RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline, because they are specific to two different groups of participants.

### 7.3.1.1. Discrete Metric Example: Number of Grip Aperture Adjustments

To understand whether participants altered their device control behaviour when using the simulated prosthesis in high limb positions, the Number of Grip Aperture Adjustments made by a participant in RCRT Down Grasp phases were analyzed (an example of which is presented in Figure 7-8). Figure 7-8A illustrates the median number of such adjustments for RCNN-Class versus LDA-Baseline and for RCNN-Reg versus LDA-Baseline, with significant differences identified only between the latter. To further understand differences between RCNN-Reg versus LDA-Baseline, grip aperture examples from the same participant using RCNN-Reg and LDA-Baseline control were analyzed (with Movement 3, which was at a high limb position and therefore where control was most difficult, presented in Figure 7-8B and C, respectively). In subplots 7B and 7C, the Grasp phases are highlighted, grip aperture adjustments in these phases are identified,

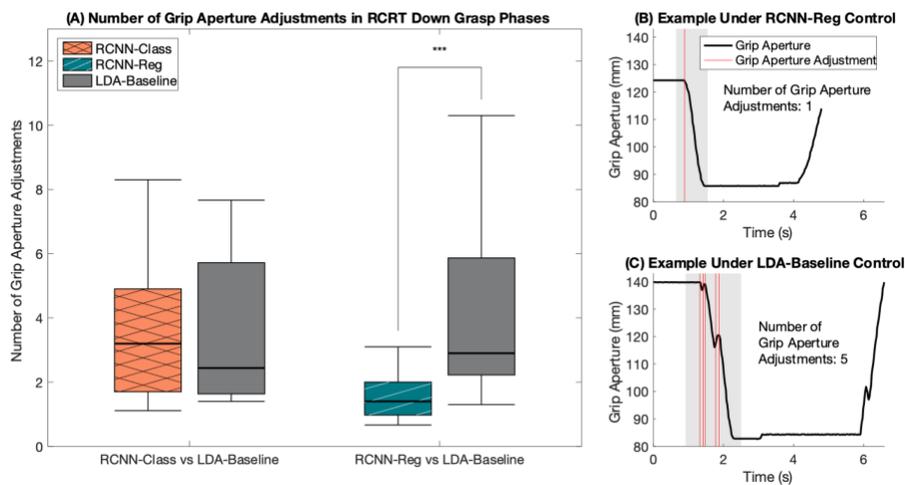


Figure 7-8. (A) Median Number of Grip Aperture Adjustments in RCRT Down Grasp phases, comparing RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline, with interquartile ranges indicated with boxes and significant differences indicated with asterisks (\*\*\*) indicates  $p < 0.05$ ); (B) grip aperture in Movement 3 of one trial under RCNN-Reg Control from a participant; and (C) grip aperture in Movement 3 of the one trial under LDA-Baseline Control from the same participant as panel B. The Grasp phases in panels B and C are highlighted in grey and each Grasp phase grip aperture adjustment is identified with a red line.

and these adjustments are tallied. Notably, only one grip aperture adjustment was identified in the RCNN-Reg example (Figure 7-8B) whereas five grip aperture adjustments were identified in the LDA-Baseline example (Figure 7-8C). Evidently fewer adjustments were required under RCNN-Reg control.

### 7.3.1.2. Continuous Metric Example: Grip Aperture Plateaus

To further understand whether participants altered their device control behaviour when using the simulated prosthesis in high limb positions, Grip Aperture Plateaus of a participant in RCRT Down Reach-Grasp movement segments were analyzed (an example of which is presented in Figure 7-9). Figure 7-9A illustrates the median plateaus for RCNN-Class versus LDA-Baseline and for RCNN-Reg versus LDA-Baseline, with significant differences identified only between the latter. To further understand differences between RCNN-Reg versus LDA-Baseline, grip aperture examples from the same participant using RCNN-Reg and LDA-Baseline control were analyzed (with Movement 3, which was at a high limb position and therefore where control was most difficult, presented in Figure 7-9B and C, respectively). In subplots 8B and 8C, Grip Aperture Plateaus are highlighted, and these plateaus are summed. Notably, only a 0.86 s plateau was identified in the

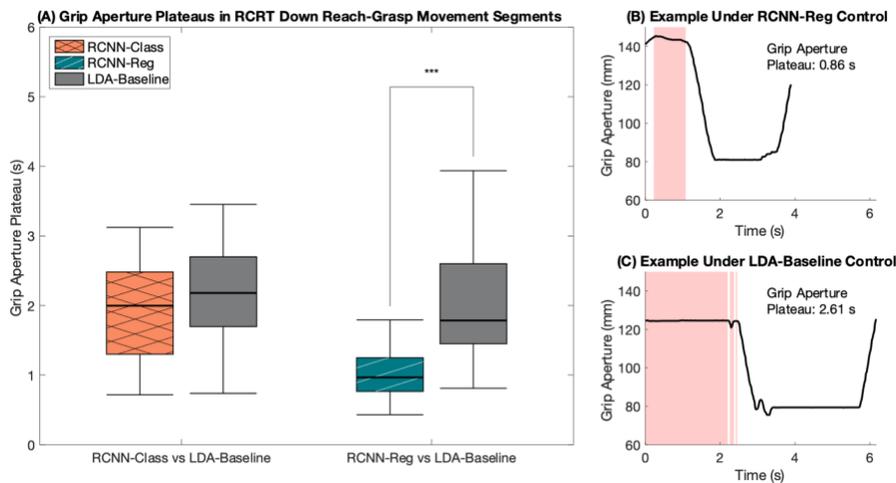


Figure 7-9. (A) Median Grip Aperture Plateaus in RCRT Down Reach-Grasp movement segments, comparing RCNN-Class versus LDA-Baseline and RCNN-Reg versus LDA-Baseline, with interquartile ranges indicated with boxes and significant differences indicated with asterisks (\*\*\*) indicates  $p < 0.05$ ); (B) grip aperture in Movement 3 of one trial under RCNN-Reg Control from a participant; and (C) grip aperture in Movement 3 of the one trial under LDA-Baseline Control from the same participant as panel B. The Grip Aperture Plateaus are identified with red shading.

RCNN-Reg example (Figure 7-9B) whereas a 2.61 s plateau was identified in the LDA-Baseline example (Figure 7-9C). Evidently more natural movements resulted under RCNN-Reg control.

### 7.3.2. Metrics-Substantiated Evidence of Limb Position Effect

**LDA-Baseline**—Under LDA-Baseline control, a total of 13 instances of the limb position effect were identified in Pasta and RCRT Down tasks' Control Characteristics metrics. During Pasta execution, the following instances confirmed the limb position effect problem: in 1 of 4 metrics in Reach, 1 of 1 metric in Reach-Grasp, 4 of 4 metrics in Grasp, 2 of 4 metrics in Transport, and 2 of 3 metrics in Release. During RCRT Down execution, 3 of 4 metrics in Grasp showed evidence of the limb position effect problem (one of which is illustrated in Figure 7-6A). Notably, the limb position effect was not identified in RCRT Up under LDA-Baseline control.

**RCNN-based Controllers**—The limb position effect was never identified under RCNN-Class or RCNN-Reg control, suggesting that both such controllers might have mitigated the limb position effect.

### 7.3.3. RCNN-Reg Simultaneous and Proportional Velocity Control Capabilities

Participants in this study took advantage of the simultaneous and proportional velocity control capabilities offered by RCNN-Reg. Figure 7-10A illustrates simultaneous control use in Reach, Grasp, and Release phases of RCNN-Reg in RCRT Up and Down tasks. Figure 7-10B illustrates proportional velocity control use of grip aperture and wrist rotation using RCNN-Reg in all three tasks.

## 7.4. Discussion

In this work, RCNN control using either classification (RCNN-Class) or regression (RCNN-Reg), versus LDA-Baseline classification control were compared. Analysis of comprehensive myoelectric control evaluation metrics [99] yielded rich, limb-position-related outcomes. What follows is a discussion about such outcomes, with a focus on understanding control challenges experienced during participants' execution of the Pasta Box Task (Pasta) and the Refined Clothespin Relocation Test (RCRT Up and RCRT Down).

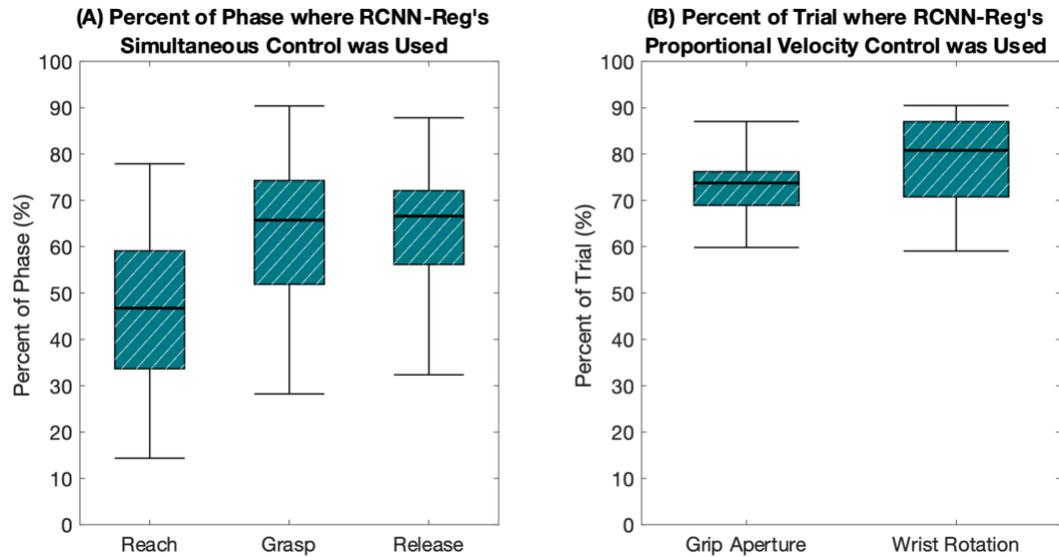


Figure 7-10.(A) The median percentage of each Reach, Grasp, and Release phase of RCRT Up and Down where grip aperture and wrist rotation were simultaneously controlled using RCNN-Reg, and (B) the median percentage of each trial where grip aperture and wrist rotation were each controlled at a velocity between no movement and maximum velocity.

The first major finding from this comparative controller research study is that RCNN-Reg offers more accurate and position-aware prosthesis control versus LDA-Baseline. This assertion is evidenced by four key findings:

- RCNN-Reg performed significantly better than LDA-Baseline in 38 control evaluation metrics.
- Fifteen of these 38 metrics were in the specific phases where instances of the limb position effect occurred (that is, in all phases of the Pasta functional task, and Grasp phases of RCRT Down task). As such, RCNN-Reg successfully mitigated these instances of the limb position effect.
- Furthermore, RCNN-Reg performed significantly better than LDA-Baseline in every phase of each functional task performed in this study. Given this, RCNN-Reg can be said to provide accurate control under a variety of limb positions.
- Finally, RCNN-Reg performed significantly better than LDA-Baseline in 12 Pasta task metrics. Recall that this task requires limb positions not included in RCNN-Reg model's training routine (that is, cross-body and away-from-body movements). This suggests that

RCNN-Reg can maintain its predictive movement accuracy in limb positions for which it is not specifically trained.

A second major takeaway from this study is that RCNN-Reg likely offers more accurate control as compared to RCNN-Class. Although RCNN-Reg and RCNN-Class were not directly compared in this work to respect participants' time, their respective number of significant improvements over LDA-Baseline can be compared to offer evidence of this conclusion:

- In the Pasta task, RCNN-Class was significantly better than LDA-Baseline in 1 metric and RCNN-Reg was significantly better than LDA-Baseline in 12 metrics → RCNN-Reg yielded 11 more control improvement instances.
- In RCRT Up, RCNN-Class was never significantly better than LDA-Baseline, whereas RCNN-Reg was significantly better than LDA-Baseline in 14 metrics → RCNN-Reg yielded 14 more control improvement instances.
- In RCRT Down, RCNN-Class was significantly better than LDA-Baseline in 10 metrics and RCNN-Reg was significantly better than LDA-Baseline in 12 metrics → RCNN-Reg yielded 2 more control improvement instances.

A third major finding from this work is that RCNN-Reg offers more natural functionality versus LDA-Baseline or even RCNN-Class. RCNN-Reg affords users the ability to control multiple device movements at a given time (such as those of the wrist and hand), and to control device movement velocity. It is recognized in the literature that regression-based myoelectric control is natural and flexible—device functions can be accessed in combination, with their velocities being independently controlled [80]. This work examined whether participants took advantage of these control capabilities, as offered by RCNN-Reg. The following two observations were uncovered:

- Participants *did* take advantage of simultaneous control. Under RCNN-Reg control, simultaneous movements during task execution were indeed evidenced (see Figure 7-10A).
- Participants also took advantage of movement velocity control capabilities. Under RCNN-Reg control, mid-velocity hand and wrist movements were evidenced during task execution (that is, velocities ranging between no movement and maximum velocity were found; see Figure 7-10B).

Despite the improved control outcome takeaways, particularly as offered by RCNN-Reg, no significant differences between controllers under investigation were identified in the NASA-TLX and usability surveys. Interestingly, though not statistically significant, participants did rate the mental demand required of RCNN-Reg to be high (see Appendix D). Mental demand had the largest difference between RCNN-Reg and LDA-Baseline scores, in comparison to all survey dimensions. This outcome aligns with the notion that progressive learning needs to be allocated for complicated hand-tasks, given that such tasks are mentally demanding [80]. As RCNN-Reg control offers both simultaneous and velocity control capabilities, it is reasonable to assume that introducing a revised and perhaps longer, progressive learning approach during controller practice sessions will improve users' appreciation of these improvements. Our earlier research, which employed the same surveys, offered other important considerations—that users' perceptions of poor or diminished control between sessions might change over time, and that reminders of earlier session control perceptions would have helped to establish first-session anchor scores [151]. This work concurs with these considerations and recommends that the Prosthesis Task Load Index (PROS-TLX) [152] be considered in future comparative prosthesis control research.

It is understood in the literature that two key factors influence user-assessments of upper limb prosthetic devices (hands in particular)—intuitive control experiences and minimal burden placed on the user (including training burden) [156], [157]. This study's RCNN-Reg offered control improvements over RCNN-Class and LDA-Baseline but required longer training routines (training routine durations of 300, 200, and 50 seconds, respectively). It is worth noting that improved control should not come at the cost of a burdensome training routine, as the latter will overshadow users' perceptions of control benefits [158]. The longer training routine required of RCNN-Reg, therefore, must not be ignored despite the users' benefits of simultaneous control over multiple degrees of freedom and movement velocity control. It is recommended that routines based on continuous hand movements be streamlined, to make model training less of a burden for users. In addition, transfer learning methods should be explored as a means of minimizing training burden introduced by RCNN-Reg [90].

This current study's RCNN-Reg success is believed to be largely due to how the training routine was implemented (despite its length) and how the prediction smoothing was undertaken. Participants followed on-screen sinusoids during training, to ensure that they executed well-paced

continuous wrist and hand motions. This on-screen method kept them engaged and focused throughout all multi-limb-position training. Routine implementation methods are important, given that successfully training a control model in multiple limb positions is known to improve prosthesis control [15], [123], [159]. Along with use of an engaging multi-limb-position training method, prediction smoothing was rigorously undertaken in this study. It is recognized that smoothing a continuous model's output, including that from RCNN-Reg, can improve model stability [160]. So together, the model training and prediction smoothing methods used in this work were key factors in RCNN-Reg's performance success, its ability to mitigate the limb position effect, and its capacity to offer simultaneous wrist/hand movements with variable velocity.

Future work should build upon the regression-based control successes uncovered in this study, including: introducing simultaneous wrist and hand movements to the model training routine; extending model training and testing to offer more degrees of freedom; averaging three or more predictions (rather than two, as used in this work) to potentially improve prediction smoothing; and investigating the window size and overlap used in data processing (given that RCNN-based models are not yet the norm in prosthesis control). Finally, to corroborate our finding that the limb position effect can indeed be mitigated, future control model training and testing should be conducted by persons with amputation.

## 7.5. Conclusion

This work contributes to upper limb myoelectric prosthesis research by offering a novel regression-based controller that employs deep learning methods. The controller provides both simultaneous control of multiple device movements at a given time and control over device movement velocity, across numerous limb positions—capabilities much closer to the fluid movements of an intact wrist and hand. In this study, an RCNN classification controller and an RCNN regression controller were each compared to a commonly used LDA classification (baseline) alternative. Of these, the regression-based approach offered the most reliable and fluid prosthetic wrist/hand movements. Offering such position-aware control, however, required a training routine that was longer than that of baseline pattern recognition, and therefore could be considered burdensome to the user. Nevertheless, regression-based solutions should continue to be investigated in future RCNN control studies, as we expect newer models to yield similarly smooth and natural wrist and hand movements. We recommend that model architectures be reimaged, training routines be

altered to include simultaneous wrist and hand movements, and that transfer learning methods be applied to minimize training burden. Given that RCNN regression-based control considers the complex and dynamic process of a wrist and hand functioning in concert with varied velocities, we recognize this as a promising direction for myoelectric upper limb device control that satisfies user needs and wants. As such, next-step testing and validation of RCNN regression-based control should include persons with transradial amputation, to confirm the validity of our proposed controller's benefits. Finally, as EMG-based controllers are not restricted to upper limb prosthesis applications, this research has far-reaching implications—towards use in rehabilitation exoskeletons and even EMG-activated video games. When the limb position effect is solved, acceptance of future device movement technologies by clinicians and users should become a reality.

# Chapter 8.

## RCNN Regression Control: Tested by an Individual with Amputation

Chapter 8 presents “Evidence that a deep learning regression-based controller mitigates the limb position effect for an individual with transradial amputation,” originally accepted for publication in the Proceedings of the Myoelectric Controls Symposium, in 2024 [161]. It presents a case study that furthers the work of Chapter 7. In this current chapter, the novel deep learning regression-based controller presented in Chapter 7 is compared to a baseline classification controller alternative. Outcomes from the deep learning regression-based controller tested present an exciting new direction for upper limb control. The novel solution offers a participant with transradial amputation smooth and simultaneous prosthesis movements with varied velocities. An illustration of how the information presented in Chapter 8 contributes to the overall work in this thesis is presented in Figure 8-1.

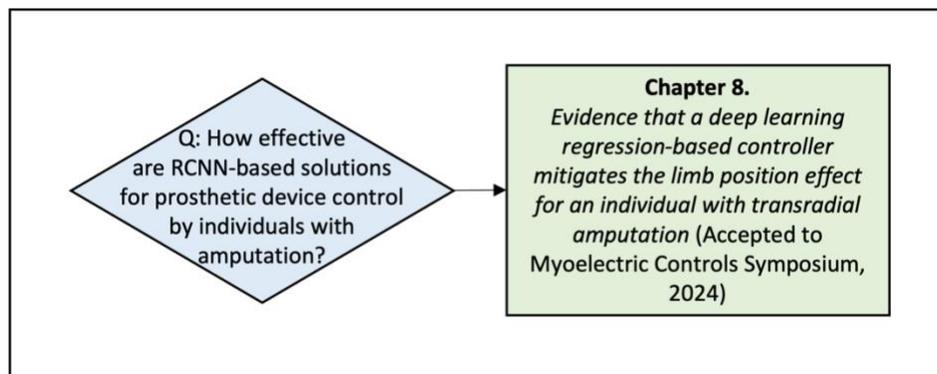


Figure 8-1. Research question addressed by Chapter 8.

## Abstract

Myoelectric upper limb prostheses provide wrist and hand movements to users yet remain somewhat unreliable and challenging to operate in high and cross-body limb positions. Hand and wrist movements are typically controlled sequentially and at a pre-set velocity. We have made significant inroads towards developing a novel controller that is reliable in multiple limb positions and offers fluid movements. Our recent work unveiled a promising *deep learning regression-based myoelectric control solution*. Herein we present results from our current study that tested device control using our solution versus a baseline (*classification*) alternative. A myoelectric prosthesis user with transradial amputation donned an experimental prosthesis and performed two functional tasks under each control option. The user exhibited superior device controllability across multiple limb positions using our regression-based solution. This work contributes evidence that a deep learning regression control approach can elicit accurate, simultaneous, and proportional device movements, while mitigating the limb position effect for a transradial prosthesis user.

### 8.1. Introduction

A myoelectric prosthesis connects to a user's residual limb via a socket and is normally controlled using electromyography (EMG) signals. EMG signals are generated by residual muscle contractions, detected by surface electrodes within the socket, and transmitted to the device's onboard controller. Pattern recognition-based controllers decode these signals to predict the user's intended movements and send corresponding commands to the device's motors. The most advanced controllers generally use a classification algorithm (model), which predicts one device action (or class) at a time [6]. The resulting prosthetic limb movements are somewhat robotic and are delivered at a pre-set velocity. That is, the wrist and hand cannot inherently move together simultaneously or with varied velocity.

Typically, individuals with transradial amputation are capable of reliably performing a limited number of residual limb muscle contractions [6]. These distinct contractions are selected to control predetermined device grasp patterns (such as hand open or close) and wrist rotation [6]. To initialize their pattern recognition-based control model for daily use, they must first perform a predetermined series of muscle contractions, known as a training routine. Through training, patterns observed in captured EMG signals are associated with corresponding device actions [7]. A

prevalent control problem, known as the *limb position effect*, results when users attempt to use their devices in untrained limb positions [14]. Oftentimes, users struggle to regain control in response to this problem. To mitigate the effect, a control model must be trained in multiple limb positions [14].

Pattern recognition-based control models can be developed using a recurrent convolutional neural network (RCNN) approach. RCNNs are a type of network architecture for *deep learning*, capable of learning directly from and handling large amounts of multimodal data [27]. These capabilities lend themselves to the capture of limb position data for improved device control. In our prior work, we merged EMG data with accelerometer data from an inertial measurement unit (EMG+IMU) using an RCNN [10], [154]. We also tested an EMG+IMU regression-based control model (RCNN-Reg) [154] and found that regression models can yield smooth device movements, given that they can predict multiple movements at once, each proportional to muscle contraction intensity and across multiple limb positions [10]. In that study, we compared RCNN-Reg to a classification-based alternative that is commonly used in control comparisons (linear discriminant analysis, LDA-Baseline) [154]. Model testing involved participants without amputation (with a simulated prosthesis donned, a reasonable proxy for a person with amputation [77]) who performed two functional tasks—the Refined Clothespin Relocation Test (RCRT) [96] and the Pasta Box Task [95]. Our work 1) found that RCNN-Reg mitigated the limb position effect; 2) substantiated that participants could perform simultaneous wrist rotation and hand open/closed movements at varied velocities, as offered by RCNN-Reg; and 3) reported that RCNN-Reg yielded better predictive accuracy than LDA-Baseline [10], [99].

This current investigation extends our earlier RCNN-Reg control model work by testing its translatability to those with upper limb loss. To accomplish this, one individual with transradial amputation donned an experimental myoelectric prosthesis that was controlled by RCNN-Reg in one session and then by LDA-Baseline in a separate session. All experimentation methods from our earlier research study ([154]) were followed. We were excited to find that RCNN-Reg indeed provided accurate, simultaneous, and proportional device movements for an individual with amputation, while mitigating the limb position effect. As such, RCNN-Reg might well offer a valuable control model alternative to traditional classification-based approaches for consideration in future myoelectric control research.

## 8.2. Methods

One participant with amputation was recruited for this study. She was female, with an age of 50 years, a height of 167 cm, and corrected-to-normal vision. She was right-handed prior to amputation. Two years prior to this study, she had a right-side transradial amputation. Her residual limb was 18 cm long with a circumference of 21 cm at the widest point. The participant typically used a myoelectric hand. She also had some at-home commercial pattern recognition-based prosthesis control experience with two degrees of freedom (wrist rotation and hand) at an estimated usage of 6 hours/day over 3 weeks. She provided written informed consent, as approved by the University of Alberta Health Research Ethics Board (Pro00086557). The participant trained and tested RCNN-Reg in her first session and LDA-Baseline in her second session, with 28 days between the two sessions. An overview of our equipment, experimentation methods, and control analysis metrics is illustrated in Figure 8-2. Full protocol details can be found in our earlier works [99], [154].

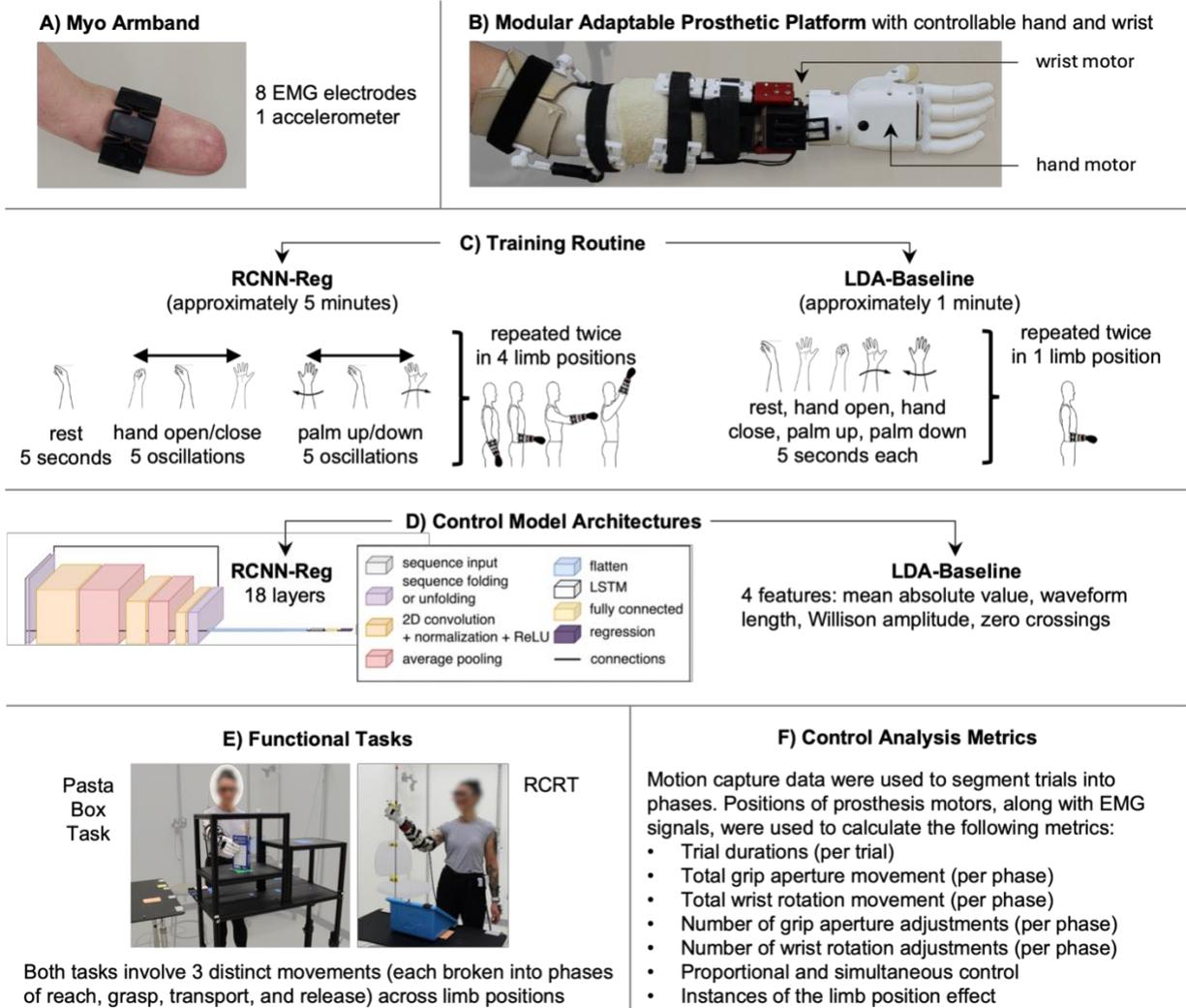


Figure 8-2. Equipment, experimentation, and analysis metrics in order of use: A) donned Myo armband, B) donned prosthesis [92], C) training routine employed for RCNN-Reg and LDA-Baseline models, including overall training durations [154], D) control model architecture details for RCNN-Reg and LDA-Baseline [154], E) functional tasks performed to test control—the Pasta Box Task [95] and the Refined Clothespin Relocation Test (RCRT) [96], split into RCRT Up and RCRT Down, and F) select metrics calculated for control analysis from our suite of metrics for comparative myoelectric prosthesis control [99].

### 8.3. Results

RCNN-Reg yielded comparatively better myoelectric prosthesis control than LDA-Baseline. RCNN-Reg’s improved control, barring a few exceptions, was evidenced by: 1) lower trial durations; 2) less total grip aperture and wrist rotation movement (indicators of corrections in mm and degrees, respectively); 3) simultaneous control of the prosthetic hand and wrist; along with 4)

fewer grip aperture and wrist rotation adjustments (number of corrections). Results are illustrated in Figure 8-3A–F, respectively, with Figure 8-3A–C,2E–F showing exceptions where *some* grip and wrist rotation corrections were made by the participant. It was also apparent that the participant took advantage of RCNN-Reg’s proportional control capabilities during task execution. That is, her hand and wrist movements occurred at a less-than-maximal velocity 100% of the time.

In our previous work [154], *under LDA-Baseline control, the limb position effect was identified a total of 13 times* (as determined through analysis of median and interquartile range trends [99]). Ten instances were evidenced during the Pasta Box Task (in 1 reach, 1 reach-grasp, 4 grasp, 2 transport, and 2 release metrics) and 3 instances during RCRT Down (all in grasp metrics). Conversely, in this present work, *under RCNN-Reg control, the limb position effect was identified a total of 3 times*, all during the Pasta Box Task (determined using our earlier work’s analysis methods [99]). Such instances were evidenced in 1 reach metric and 2 transport metrics.

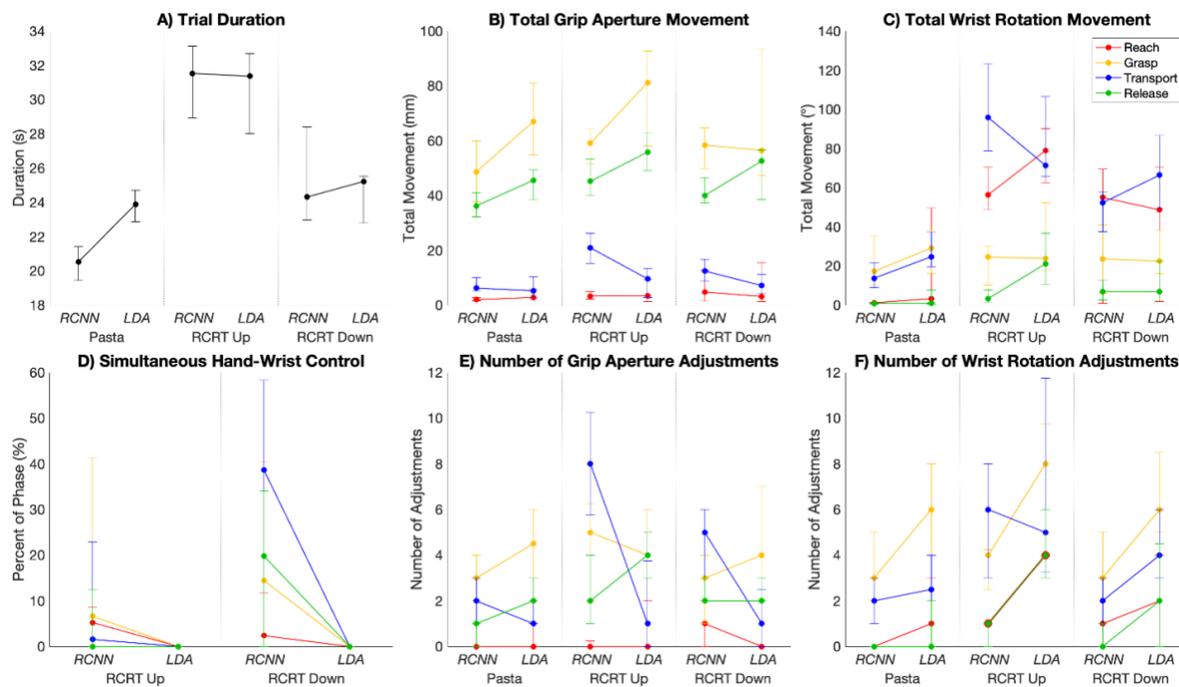


Figure 8-3. Controller comparison results (abbreviated as RCNN/LDA), including median A) trial duration, B) total grip aperture movement, C) total wrist rotation movement, D) simultaneous hand-wrist control, E) number of grip aperture adjustments, and F) number of wrist rotation adjustments. Medians are presented for each task, and for each phase (Reach in red, Grasp in orange, Transport in blue, and Release in green) where applicable. Interquartile ranges across trials are indicated with coloured error bars where applicable. Note that simultaneous hand-wrist control is only illustrated for RCRT Up/Down, as wrist rotation is not required in the Pasta Box Task (Pasta).

## 8.4. Discussion & Future Work

***Pasta Box Task Control Successes and Challenges:*** Although RCNN-Reg control was generally superior to LDA-Baseline, our participant exhibited some unexpected control challenges in the transport phases of the Pasta Box Task (Figure 8-3B,E’s abovementioned control exceptions)—evidenced by increased total grip aperture movement and grip aperture adjustments. The Pasta Box Task requires a participant to transport a box in both cross-body and away-from-body limb positions (the specific instances where the limb position effect under RCNN-Reg control occurred). Notably, RCNN-Reg’s training routine did *not* include these specific limb positions. Given these circumstances, the addition of cross-body and away-from-body limb positions to RCNN-Reg’s training routine is advised. Future work should investigate which training positions improve device control.

***RCRT Up and Down Control Successes and Challenges:*** When testing RCNN-Reg, our participant did not experience the limb position effect during execution of RCRT Up and Down. The effect was likely mitigated because RCNN-Reg’s control model was trained in the same high limb positions required of these tasks. Despite this, the participant exhibited control challenges in transport phases of RCRT Up and Down (Figure 8-3B,C,E,F’s abovementioned control exceptions)—evidenced by increased total grip aperture and wrist rotation movement, along with more grip aperture and wrist rotation adjustments, versus under LDA-Baseline control. These outcomes indicated her need to perform movement corrections to maintain control. RCRT’s transport phases require a participant to rotate their wrist while not dropping the grasped clothespin, an inherently complicated movement. Although our participant exhibited challenges, she *did* make use of the simultaneous control capabilities offered by RCNN-Reg in transport phases (Figure 8-3D). As progressive learning needs to be allocated for complicated hand-tasks [80], we believe she could master simultaneous control and exhibit fewer adjustments in RCRT’s transport phases if afforded more controller-use practice.

***Training Routine Implications:*** RCNN-Reg’s 5-minute training routine might prove to be burdensome for a user if repeated multiple times throughout the day (for re-calibration purposes). Future work should be undertaken to shorten the training routine duration. For instance, reducing

the number of oscillations performed might be feasible. The incorporation of dynamic limb positions might also reduce routine duration *and* provide richer training data [66].

***Future Experimentation:*** We recognize that our work did not exhaustively investigate RCNN-Reg. To corroborate our findings, the experimentation must be repeated by more participants with transradial amputation.

## 8.5. Conclusion

Our case study contributes, for the first time, evidence that an RCNN regression-based controller offers a user with transradial amputation control of their prosthetic wrist and hand simultaneously across multiple limb positions, using varied velocities. *RCNN-Reg successfully mitigated the limb position effect during device use*, given that our participant was able to retain control of their myoelectric prosthesis throughout numerous re-orientations of their limb in space during different tasks. Although regression-based control solutions have not garnered as much research attention as classification-based counterparts, the merits of RCNN-Reg, as presented in this work, suggest that it be strongly considered as a future prosthesis controller.

# Chapter 9.

## Conclusions and Future Directions

To the best of our knowledge, this thesis offers the first myoelectric prosthesis controller that successfully mitigates the limb position effect *and* provides users with simultaneous control over hand and wrist movements at varied velocities. Widespread acceptance of such methods for pattern recognition-based control of upper limb prostheses should *ideally* become a reality. Until this occurs, it is expected that state-of-the-art devices will remain challenging for users to control and will offer unnatural movement qualities.

Collectively, the work presented in this thesis forms a foundation that will see translation of a novel deep learning regression-based myoelectric control solution beyond the laboratory, for use in advanced prosthesis controllers. All such work focused on methods to mitigate the limb position effect. Firstly, the control solution presented is based on the development of a **position-aware RCNN regression-based model**. Secondly, its associated training routine teaches the model about **dynamic wrist and hand movements in varied limb positions**. Thirdly, **methods to reduce training burden** were explored, with actionable approaches identified. Finally, the **metrics needed to assess control across multiple limb positions** were established so that future work undertaken to solve the limb position effect problem can be substantiated. Although regression controllers have not garnered as much research attention as classification counterparts, the merits of the RCNN regression-based control solution presented in this thesis suggest that it be strongly considered for future myoelectric prosthesis control.

Figure 9-1 identifies future work to build upon the early RCNN regression-based control solution presented in this thesis. Here, three gap statements stem from the outcomes of Chapter 8, suggesting that: (1) RCNN regression-based control be assessed using a substantial number of individuals with amputation to yield statistically robust results—that is, valid results across varied conditions; (2) the model’s training routine be revised to reduce user training burden (possibly by implementing transfer learning techniques as explored in Chapter 5 and Chapter 6 of this thesis,

or a composite model method detailed in Chapter 4), and (3) the model’s training routine be augmented with new cross-body and away-from-body limb positions to further mitigate the limb position effect. In the end, a refined and rigorously tested RCNN regression-based control solution is expected to improve overall myoelectric prosthesis user satisfaction, and thereby decrease device abandonment.

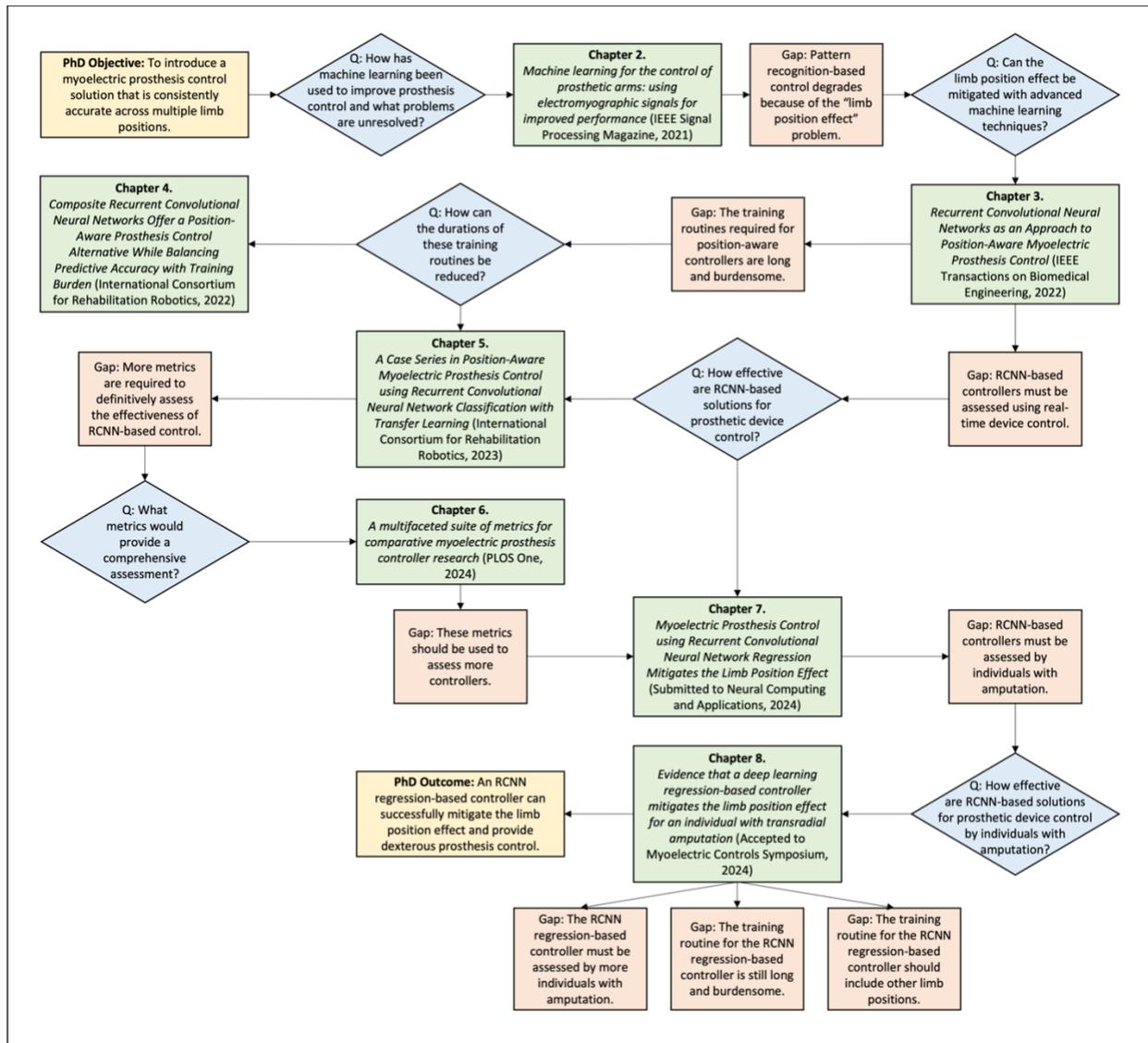


Figure 9-1. Summary of thesis chapters, with this work’s main objective and its outcome in yellow, research questions addressed in blue, chapter titles with associated publications in green, and intermittent research gaps in orange. Finally, gaps for future work are presented, each resulting from the work of Chapter 8.

Not only will the work undertaken in this thesis advance the myoelectric control field, but a final/refined RCNN regression-based model solution is expected to be translatable to several wearable rehabilitation technologies. For instance, its method to combine EMG and IMU data could be applied to other rehabilitation technologies that would similarly benefit from richer movement datasets. Examples include intelligent lower prosthetic limbs, exoskeletons that aid in spinal cord injury rehabilitation, and exoskeletons that augment muscle capacity to reduce user discomfort and fatigue. The same data combination methods might also be applicable to the latest in human movement wearables, including those for sport biomechanics outcome predictions and those used to learn and predict player movements to make video game content more dynamic and compelling. In these applications, optimal movement patterns might be more easily identified and/or predicted when using combined EMG and IMU data and deep neural networks. Finally, given that the number of people living with limb loss of any kind is expected to increase [162], this work's novel deep learning regression-based approach for myoelectric prosthesis control has come at a most opportune time—providing evidence for a new approach towards solving a user's need for smooth, reliable, and robust device operation.

# References

- [1] C. L. McDonald et al., “Global prevalence of traumatic non-fatal limb amputation,” *Prosthet Orthot Int*, vol. 45, no. 2, pp. 105–14, 2020, doi: 10.1177/0309364620972258.
- [2] C. T. Mai et al., “National population-based estimates for major birth defects, 2010–2014,” *Birth Defects Res*, vol. 111, no. 18, pp. 1420–35, 2019, doi: 10.1002/bdr2.1589.
- [3] S. E. Parker et al., “Updated national birth prevalence estimates for selected birth defects in the United States, 2004-2006,” *Birth Defects Res A Clin Mol Teratol*, vol. 88, no. 12, pp. 1008–16, 2010, doi: 10.1002/bdra.20735.
- [4] A. W. Franzke et al., “Users’ and therapists’ perceptions of myoelectric multi-function upper limb prostheses with conventional and pattern recognition control,” *PLOS One*, vol. 14, no. 8, p. e0220899, 2019, doi: 10.1371/journal.pone.0220899.
- [5] S. Salminger et al., “Current rates of prosthetic usage in upper-limb amputees—have innovations had an impact on device acceptance?,” *Disabil Rehabil*, vol. 44, no. 14, pp. 3708–13, 2022, doi: 10.1080/09638288.2020.1866684.
- [6] P. Geethanjali, “Myoelectric control of prosthetic hands: State-of-the-art review,” *Med Devices (Auckl)*, vol. 9, pp. 247–55, 2016, doi: 10.2147/MDER.S91102.
- [7] E. Scheme and K. Englehart, “Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use,” *J Rehabil Res Dev*, vol. 48, no. 6, pp. 643–59, 2011, doi: 10.1682/Jrrd.2010.09.0177.
- [8] M. M. Lalu et al., “Identifying and understanding factors that affect the translation of therapies from the laboratory to patients: A study protocol,” *F1000Res*, vol. 9, p. 485, 2020, doi: 10.12688/f1000research.23663.2.
- [9] N. Parajuli et al., “Real-time EMG based pattern recognition control challenges and future implementation,” *Sensors (Switzerland)*, vol. 19, no. 20, p. 4596, 2019, doi: 10.3390/s19204596.

- [10] H. E. Williams et al., “Recurrent convolutional neural networks as an approach to position-aware myoelectric prosthesis control,” *IEEE Trans Biomed Eng*, vol. 69, no. 7, pp. 2243–55, 2022, doi: 10.1109/TBME.2022.3140269.
- [11] H. E. Williams et al., “Composite recurrent convolutional neural networks offer a position-aware prosthesis control alternative while balancing predictive accuracy with training burden,” *Proc Int Conf Rehabil Robot (ICORR)*, 2022, doi: 10.1109/ICORR55369.2022.9896593.
- [12] F. Cordella et al., “Literature review on needs of upper limb prosthesis users,” *Front Neurosci*, vol. 10, p. 209, 2016, doi: 10.3389/fnins.2016.00209.
- [13] A. W. Shehata and H. E. Williams et al., “Machine learning for the control of prosthetic arms: Using electromyographic signals for improved performance,” *IEEE Signal Process Mag*, vol. 38, no. 4, pp. 46–53, 2021, doi: 10.1109/MSP.2021.3075931.
- [14] E. Campbell et al., “Current trends and confounding factors in myoelectric control: Limb position and contraction intensity,” *Sensors (Switzerland)*, vol. 20, no. 6, p. 1613, 2020, doi: 10.3390/s20061613.
- [15] A. Fougner et al., “Resolving the limb position effect in myoelectric pattern recognition,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 19, no. 6, pp. 644–51, 2011, doi: 10.1109/TNSRE.2011.2163529.
- [16] A. Radmand et al., “A characterization of the effect of limb position on EMG features to guide the development of effective prosthetic control schemes,” *Proc 36<sup>th</sup> Annu Int Conf IEEE Eng Med Biol Soc (EMBC)*, pp. 662–7, 2014, doi: 10.1109/EMBC.2014.6943678.
- [17] A. Radmand et al., “On the suitability of integrating accelerometry data with electromyography signals for resolving the effect of changes in limb position during dynamic limb movement,” *J Prosthet Orthot*, vol. 26, no. 4, pp. 185–93, 2014, doi: 10.1097/JPO.0000000000000041.

- [18] E. Scheme et al., “Examining the adverse effects of limb position on pattern recognition based myoelectric control,” *Proc Annu Int Conf IEEE Eng Med Biol (EMBC)*, pp. 6337–40, 2010, doi: 10.1109/IEMBS.2010.5627638.
- [19] E. Scheme et al., “Improving myoelectric pattern recognition positional robustness using advanced training protocols,” *Proc Annu Int Conf IEEE Eng Med Biol Soc (EMBC)*, pp. 4828–31, 2011, doi: 10.1109/IEMBS.2011.6091196.
- [20] A. Boschmann and M. Platzner, “Reducing the limb position effect in pattern recognition based myoelectric control using a high density electrode array,” *Proc ISSNIP Biosic Biorobot Conf (BRC)*, pp. 1–5, 2013, doi: 10.1109/BRC.2013.6487548.
- [21] W. Shahzad et al., “Enhanced performance for multi-forearm movement decoding using hybrid IMU–SEMG interface,” *Front Neurobot*, vol. 13, p. 43, 2019, doi: 10.3389/fnbot.2019.00043.
- [22] Y. Geng et al., “Toward attenuating the impact of arm positions on electromyography pattern-recognition based motion classification in transradial amputees,” *J Neuroeng Rehabil*, vol. 9, p. 74, 2012, doi: 10.1186/1743-0003-9-74.
- [23] Y. Geng et al., “Improving the robustness of real-time myoelectric pattern recognition against arm position changes in transradial amputees,” *Biomed Res Int*, vol. 2017, p. 5090454, 2017, doi: 10.1155/2017/5090454.
- [24] C. Igual et al., “Myoelectric control for upper limb prostheses,” *Electronics (Switzerland)*, vol. 8, no. 11, p. 1244, 2019, doi: 10.3390/electronics8111244.
- [25] C. Ahmadizadeh et al., “Human machine interfaces in upper-limb prosthesis control: A survey of techniques for preprocessing and processing of biosignals,” *IEEE Signal Process Mag*, vol. 38, no. 4, pp. 12–22, 2021, doi: 10.1109/MSP.2021.3057042.
- [26] F. Leone, “Pattern recognition algorithms for upper-limb prosthetics control,” PhD Thesis, Università Campus Bio-Medico, 2021.
- [27] A. Phinyomark and E. Scheme, “EMG pattern recognition in the era of big data and deep learning,” *Big Data Cogn Comput*, vol. 2, no. 3, p. 21, 2018, doi: 10.3390/bdcc2030021.

- [28] W. Wang et al., "Sensor fusion for myoelectric control based on deep learning with recurrent convolutional neural networks," *Artif Organs*, vol. 42, no. 9, pp. E272-82, 2018, doi: 10.1111/aor.13153.
- [29] I. Kyranou et al., "Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses," *Front Neurobot*, vol. 12, p. 58, 2018, doi: 10.3389/fnbot.2018.00058.
- [30] A. D. Roche et al., "Prosthetic myoelectric control strategies: A clinical perspective," *Curr Surg Rep*, vol. 2, p. 44, 2014, doi: 10.1007/s40137-013-0044-8.
- [31] A. D. Roche et al., "Clinical perspectives in upper limb prostheses: An update," *Curr Surg Rep*, vol. 7, p. 5, 2019, doi: 10.1007/s40137-019-0227-z.
- [32] C. J. De Luca et al., "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *J Biomech*, vol. 43, no. 8, pp. 1573-9, 2010, doi: 10.1016/j.jbiomech.2010.01.027.
- [33] R. K. Ranjan et al., "Comb filter for elimination of unwanted power line interference in biomedical signal," *J Circuit Syst Comp*, vol. 25, no. 6, p. 1650052, 2016, doi: 10.1142/S0218126616500523.
- [34] A. Phinyomark et al., "Feature reduction and selection for EMG signal classification," *Expert Syst Appl*, vol. 39, no. 8, pp. 7420-31, 2012, doi: 10.1016/j.eswa.2012.01.102.
- [35] T. R. Farrell and R. F. ff. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Trans Neural Syst Rehabil Eng*, vol. 15, no. 1, pp. 111-8, 2007, doi: 10.1109/Tnsre.2007.891391.
- [36] R. O. Duda et al., *Pattern Classification*, 2<sup>nd</sup> ed., Wiley, 2001.
- [37] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE Trans Biomed Eng*, vol. 55, no. 8, pp. 1956-65, 2008, doi: 10.1109/TBME.2008.919734.

- [38] A. Dellacasa Bellingegni et al., “NLR, MLP, SVM, and LDA: A comparative analysis on EMG data from people with trans-radial amputation,” *J Neuroeng Rehabil*, vol. 14, p. 82, 2017, doi: 10.1186/s12984-017-0290-6.
- [39] M. F. Kelly et al., “Neural network classification of myoelectric signal for prosthesis control,” *J Electromyogr Kines*, vol. 1, no. 4, pp. 229–36, 1991, doi: 10.1016/1050-6411(91)90009-T.
- [40] M. Ortiz-Catalan et al., “Real-time and simultaneous control of artificial limbs based on pattern recognition algorithms,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 4, pp. 756–64, 2014, doi: 10.1109/TNSRE.2014.2305097.
- [41] E. C. Alexopoulos, “Introduction to multivariate regression analysis,” *Hippokratia*, vol. 14, no. Suppl 1, pp. 23–8, 2010.
- [42] A. Ameri et al., “Support vector regression for improved real-time, simultaneous myoelectric control,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 6, pp. 1198–209, 2014, doi: 10.1109/TNSRE.2014.2323576.
- [43] A. Ameri et al., “Regression convolutional neural network for improved simultaneous EMG control,” *J Neural Eng*, vol. 16, no. 3, p. 036015, 2019, doi: 10.1088/1741-2552/ab0e2e.
- [44] A. W. Shehata et al., “Evaluating internal model strength and performance of myoelectric prosthesis control strategies,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, no. 5, pp. 1046–55, 2018, doi: 10.1109/TNSRE.2018.2826981.
- [45] A. L. Edwards et al., “Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching,” *Prosthet Orthot Int*, vol. 40, no. 5, pp. 573–81, 2016, doi: 10.1177/0309364615605373.
- [46] A. L. Edwards et al., “Machine learning and unlearning to autonomously switch between the functions of a myoelectric arm,” *Proc 6<sup>th</sup> IEEE Int Conf Biomed Robot Biomechatron (BioRob)*, pp. 514–21, 2016, doi: 10.1109/BIOROB.2016.7523678.

- [47] M. Merad et al., “Assessment of an automatic prosthetic elbow control strategy using residual limb motion for transhumeral amputated individuals with socket or osseointegrated prostheses,” *IEEE Trans Med Robot Bionics*, vol. 2, no. 1, pp. 38–49, 2020, doi: 10.1109/tmr.2020.2970065.
- [48] D. A. Bennett and M. Goldfarb, “IMU-based wrist rotation control of a transradial myoelectric prosthesis,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 26, no. 2, pp. 419–27, 2018, doi: 10.1109/TNSRE.2017.2682642.
- [49] S. Mick et al., “Shoulder kinematics plus contextual target information enable control of multiple distal joints of a simulated prosthetic arm and hand,” *J Neuroeng Rehabil*, vol. 18, p. 3, 2021, doi: 10.1186/s12984-020-00793-0.
- [50] A. Akhtara et al., “Estimation of distal arm joint angles from EMG and shoulder orientation for transhumeral prostheses,” *J Electromyogr Kines*, vol. 35, pp. 86–94, 2017, doi: 10.1016/j.jelekin.2017.06.001.Estimation.
- [51] J. W. Sensinger and S. Dosen, “A review of sensory feedback in upper-limb prostheses from the perspective of human motor control,” *Front Neurosci*, vol. 14, p. 345, 2020, doi: 10.3389/fnins.2020.00345.
- [52] A. W. Shehata et al., “Audible feedback improves internal model strength and performance of myoelectric prosthesis control,” *Sci Rep*, vol. 8, p. 8541, 2018, doi: 10.1038/s41598-018-26810-w.
- [53] L. F. Engels et al., “When less is more – Discrete tactile feedback dominates continuous audio biofeedback in the integrated percept while controlling a myoelectric prosthetic hand,” *Front Neurosci*, vol. 13, p. 578, 2019, doi: 10.3389/fnins.2019.00578.
- [54] A. Saudabayev and H. A. Varol, “Sensors for robotic hands: A survey of state of the art,” *IEEE Access*, vol. 3, pp. 1765–82, 2015, doi: 10.1109/ACCESS.2015.2482543.
- [55] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–33, 2015, doi: 10.1038/nature14236.

- [56] H. A. Dewald et al., “Stable, three degree-of-freedom myoelectric prosthetic control via chronic bipolar intramuscular electrodes: A case study,” *J Neuroeng Rehabil*, vol. 16, p. 147, 2019, doi: 10.1186/s12984-019-0607-8.
- [57] M. Ortiz-Catalan et al., “An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs,” *Sci Transl Med*, vol. 6, no. 257, p. 257re6, 2014, doi: 10.1126/scitranslmed.3008933.
- [58] A. Boschmann and M. Platzner, “Towards robust HD EMG pattern recognition: Reducing electrode displacement effect using structural similarity,” *Proc 36<sup>th</sup> Annu Int Conf IEEE Eng Med Biol Soc (EMBC)*, pp. 4547–50, 2014, doi: 10.1109/EMBC.2014.6944635.
- [59] A. Radmand et al., “High-density force myography: A possible alternative for upper-limb prosthetic control,” *J Rehabil Res Dev*, vol. 53, no. 4, pp. 443–56, 2016, doi: 10.1682/JRRD.2015.03.0041.
- [60] R. N. Khushaba et al., “Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features,” *Neural Networks*, vol. 55, pp. 42–58, 2014, doi: 10.1016/j.neunet.2014.03.010.
- [61] K.-H. Park et al., “Position-independent decoding of movement intention for proportional myoelectric interfaces,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 24, no. 9, pp. 928–39, 2016, doi: 10.1109/TNSRE.2015.2481461.
- [62] Y. Yu et al., “Attenuating the impact of limb position on surface EMG pattern recognition using a mixed-LDA classifier,” *Proc IEEE Int Conf Robot Biomim (ROBIO)*, pp. 1497–502, 2017, doi: 10.1109/ROBIO.2017.8324629.
- [63] R. J. Beaulieu et al., “Multi-position training improves robustness of pattern recognition and reduces limb-position effect in prosthetic control,” *J Prosthet Orthot*, vol. 29, no. 2, pp. 54–62, 2017, doi: 10.1097/JPO.000000000000121.
- [64] J. L. Betthausen et al., “Limb position tolerant pattern recognition for myoelectric prosthesis control with adaptive sparse representations from extreme learning,” *IEEE Trans Biomed Eng*, vol. 65, no. 4, pp. 770–8, 2018, doi: 10.1109/TBME.2017.2719400.

- [65] A. K. Mukhopadhyay and S. Samui, "An experimental study on upper limb position invariant EMG signal classification based on deep neural network," *Biomed Signal Process Control*, vol. 55, p. 101669, 2020, doi: 10.1016/j.bspc.2019.101669.
- [66] A. Gigli et al., "Natural myocontrol in a realistic setting: A comparison between static and dynamic data acquisition," *Proc IEEE 16<sup>th</sup> Int Conf Rehabil Robot (ICORR)*, pp. 1061–6, 2019, doi: 10.1109/ICORR.2019.8779364.
- [67] S. Asht and R. Dass, "Pattern recognition techniques: A review," *Int J Comp Sci Telecomm*, vol. 3, no. 8, pp. 25–9, 2012.
- [68] A. Phinyomark et al., "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Expert Syst Appl*, vol. 40, no. 12, pp. 4832–40, 2013, doi: 10.1016/j.eswa.2013.02.023.
- [69] A. Phinyomark et al., "Feature extraction and selection for myoelectric control based on wearable EMG sensors," *Sensors (Switzerland)*, vol. 18, no. 5, p. 1615, 2018, doi: 10.3390/s18051615.
- [70] M. Atzori et al., "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Front Neurobot*, vol. 10, p. 9, 2016, doi: 10.3389/fnbot.2016.00009.
- [71] U. Côté-Allard et al., "Interpreting deep learning features for myoelectric control: A comparison with handcrafted features," *Front Bioeng Biotechnol*, vol. 8, p. 158, 2020, doi: 10.3389/fbioe.2020.00158.
- [72] P. Xia et al., "EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks," *Artif Organs*, vol. 42, no. 5, pp. E67–77, 2017, doi: 10.1111/aor.13004.
- [73] T. Bao et al., "A CNN-LSTM hybrid framework for wrist kinematics estimation using surface," *IEEE Trans Instrum Meas*, vol. 70, pp. 88–94, 2020, doi: 10.1109/TIM.2020.3036654.

- [74] D. J. Atkins et al., "Epidemiologic overview of individuals with upper-limb loss and their reported research priorities," *J Prosthet Orthot*, vol. 8, no. 1, pp. 2–11, 1996, doi: 10.1097/00008526-199600810-00003.
- [75] A. W. Shehata et al., "Mechanotactile sensory feedback improves embodiment of a prosthetic hand during active use," *Front Neurosci*, vol. 14, p. 263, 2020, doi: 10.3389/fnins.2020.00263.
- [76] J. V. V. Parr et al., "Visual attention, EEG alpha power and T7-Fz connectivity are implicated in prosthetic hand control and can be optimized through gaze training," *J Neuroeng Rehabil*, vol. 16, p. 52, 2019, doi: 10.1186/s12984-019-0524-x.
- [77] H. E. Williams et al., "Myoelectric prosthesis users and non-disabled individuals wearing a simulated prosthesis exhibit similar compensatory movement strategies," *J Neuroeng Rehabil*, vol. 18, p. 72, 2021, doi: 10.1186/s12984-021-00855-x.
- [78] A. LeNail, "NN-SVG: Publication-ready neural network architecture schematics," *J Open Source Softw*, vol. 4, no. 33, p. 747, 2019, doi: 10.21105/joss.00747.
- [79] H.-J. Hwang et al., "Real-time robustness evaluation of regression based myoelectric control against arm position change and donning/doffing," *PLOS One*, vol. 12, no. 11, p. e0186318, 2017, doi: 10.1371/journal.pone.0186318.
- [80] J. M. Hahne et al., "User adaptation in myoelectric man-machine interfaces," *Sci Rep*, vol. 7, p. 4437, 2017, doi: 10.1038/s41598-017-04255-x.
- [81] T. A. Kuiken et al., "A comparison of pattern recognition control and direct control of a multiple degree-of-freedom transradial prosthesis," *IEEE J Transl Eng Health Med*, vol. 4, p. 2100508, 2016, doi: 10.1109/JTEHM.2016.2616123.
- [82] S. Tam et al., "Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning," *Sci Rep*, vol. 11, p. 11275, 2021, doi: 10.1038/s41598-021-90688-4.

- [83] L. H. Smith et al., “Use of probabilistic weights to enhance linear regression myoelectric control,” *J Neural Eng*, vol. 12, no. 6, p. 066030, 2015, doi: 10.1088/1741-2560/12/6/066030.
- [84] L. H. Smith et al., “Evaluation of linear regression simultaneous myoelectric control using intramuscular EMG,” *IEEE Trans Biomed Eng*, vol. 63, no. 4, pp. 737–46, 2016, doi: 10.1109/TBME.2015.2469741.
- [85] J. A. George et al., “Inexpensive surface electromyography sleeve with consistent electrode placement enables dexterous and stable prosthetic control through deep learning,” *Proc Myoelect Controls Symp (MEC20)*, pp. 103–6, 2020.
- [86] N. Malesevic et al., “A database of multi-channel intramuscular electromyogram signals during isometric hand muscles contractions,” *Sci Data*, vol. 7, p. 10, 2020, doi: 10.1038/s41597-019-0335-8.
- [87] L. J. Resnik et al., “User experience of controlling the DEKA arm with EMG pattern recognition,” *PLOS One*, vol. 13, no. 9, p. e0203987, 2018, doi: 10.1371/journal.pone.0203987.
- [88] A. M. Simon et al., “Patient training for functional use of pattern recognition-controlled prostheses,” *J Prosthet Orthot*, vol. 24, no. 2, pp. 56–64, 2012, doi: 10.1097/JPO.0b013e3182515437.
- [89] L. J. Hargrove et al., “Myoelectric pattern recognition outperforms direct control for transhumeral amputees with targeted muscle reinnervation: A randomized clinical trial,” *Sci Rep*, vol. 7, p. 13840, 2017, doi: 10.1038/s41598-017-14386-w.
- [90] H. E. Williams et al., “A case series in position-aware myoelectric prosthesis control using recurrent convolutional neural network classification with transfer learning,” *Proc Int Conf Rehabil Robot (ICORR)*, 2023, doi: 10.1109/ICORR58425.2023.10304787.
- [91] H. E. Williams et al., “Using transfer learning to reduce the burden of training for position-aware myoelectric prosthetic control,” *Proc RehabWeek Virtual ‘21*, p. P-130, 2021.

- [92] B. W. Hallworth et al., “A transradial modular adaptable platform for evaluating prosthetic feedback and control strategies,” *Proc Myoelect Controls Symp (MEC20)*, pp. 203–6, 2020.
- [93] E. D. Wells et al., “Preliminary evaluation of the effect of mechanotactile feedback location on myoelectric prosthesis performance using a sensorized prosthetic hand,” *Sensors (Switzerland)*, vol. 22, no. 10, p. 3892, 2022, doi: 10.3390/s22103892.
- [94] M. R. Dawson et al., “BrachIOplexus: Myoelectric training software for clinical and research applications,” *Proc Myoelect Controls Symp (MEC20)*, pp. 87–90, 2020.
- [95] A. M. Valevicius et al., “Characterization of normative hand movements during two functional upper limb tasks,” *PLOS One*, vol. 13, no. 6, p. e0199549, 2018, doi: 10.1371/journal.pone.0199549.
- [96] A. Hussaini and P. Kyberd, “Refined clothespin relocation test and assessment of motion,” *Prosthet Orthot Int*, vol. 41, no. 3, pp. 294–302, 2017, doi: 10.1177/0309364616660250.
- [97] C. L. Hunt et al., “Limb loading enhances skill transfer between augmented and physical reality tasks during limb loss rehabilitation,” *J Neuroeng Rehabil*, vol. 20, p. 16, 2023, doi: 10.1186/s12984-023-01136-5.
- [98] H. Bouwsema et al., “Determining skill level in myoelectric prosthesis use with multiple outcome measures,” *J Rehabil Res Dev*, vol. 49, no. 9, pp. 1331–48, 2012, doi: 10.1682/jrrd.2011.09.0179.
- [99] H. E. Williams et al., “A multifaceted suite of metrics for comparative myoelectric prosthesis controller research,” *PLOS One*, vol. 9, no. 5, p. e0291279, 2024, doi: 10.1101/2023.08.28.554998.
- [100] D. W. Braza and J. N. Yacub Martin, “Upper limb amputations,” in *Essentials of Physical Medicine and Rehabilitation: Musculoskeletal Disorders, Pain, and Rehabilitation*, 4<sup>th</sup> ed., Elsevier Inc., pp. 651–7, 2018, doi: 10.1016/B978-0-323-54947-9.00119-X.

- [101] L. H. Smith et al., “Myoelectric control system and task-specific characteristics affect voluntary use of simultaneous control,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 24, no. 1, pp. 109–16, 2016, doi: 10.1109/TNSRE.2015.2410755.
- [102] Y. Du et al., “Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation,” *Sensors (Switzerland)*, vol. 17, no. 3, p. 458, 2017, doi: 10.3390/s17030458.
- [103] E. F. Shair et al., “Reduction of limb position invariant of SEMG signals for improved prosthetic control using spectrogram,” *Int J Integr Eng*, vol. 13, no. 5, pp. 79–88, 2021, doi: 10.30880/ijie.2021.13.05.010.
- [104] P. K. Koppolu and K. Chemmangat, “A two-stage classification strategy to reduce the effect of wrist orientation in surface myoelectric pattern recognition,” *Proc IEEE Int Conf Sig Proces Comm (SPCOM)*, 2022, doi: 10.1109/SPCOM55316.2022.9840809.
- [105] J. S. Gharibo and M. D. Naish, “Multi-modal prosthesis control using sEMG, FMG and IMU sensors,” *Proc 44<sup>th</sup> Annu Int Conf IEEE Eng Med Biol Soc (EMBC)*, pp. 2983–7, 2022, doi: 10.1109/EMBC48229.2022.9871586.
- [106] V. Spieker et al., “An adaptive multi-modal control strategy to attenuate the limb position effect in myoelectric pattern recognition,” *Sensors (Switzerland)*, vol. 21, no. 21, p. 7404, 2021, doi: 10.3390/s21217404.
- [107] H. E. Williams et al., “Hand function kinematics when using a simulated myoelectric prosthesis,” *Proc IEEE 16<sup>th</sup> Int Conf Rehabil Robot (ICORR)*, Toronto, Canada, pp. 169–74, 2019, doi: 10.1109/ICORR.2019.8779443.
- [108] Q. A. Boser, “Characterizing the visuomotor behaviour of upper limb body-powered prosthesis users,” MSc Thesis, University of Alberta, 2019.
- [109] H. E. Williams et al., “Gaze and movement assessment (GaMA): Inter-site validation of a visuomotor upper limb functional protocol,” *PLOS One*, vol. 14, no. 12, p. e0219333, 2019, doi: 10.1371/journal.pone.0219333.

- [110] V. Mathiowetz et al., “Adult norms for the box and block test of manual dexterity,” *Am J Occup Ther*, vol. 39, no. 6, pp. 386–91, 1985, doi: 10.5014/ajot.39.6.386.
- [111] R. H. Jebsen et al., “An objective and standardized test of hand function,” *Arch Phys Med Rehabil*, vol. 50, no. 6, pp. 311–9, 1969.
- [112] L. Resnik et al., “Development and evaluation of the activities measure for upper limb amputees,” *Arch Phys Med Rehabil*, vol. 94, no. 3, pp. 488–94.e4, 2013, doi: 10.1016/j.apmr.2012.10.004.
- [113] C. M. Light et al., “Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity,” *Arch Phys Med Rehabil*, vol. 83, no. 6, pp. 776–83, 2002, doi: 10.1053/apmr.2002.32737.
- [114] L. M. Hermansson et al., “Assessmet of capacity for myoelectric control: A new rasch-built measure of prosthetic hand control,” *J Rehabil Med*, vol. 37, no. 3, pp. 166–71, 2005, doi: 10.1080/16501970410024280.
- [115] P. Capsi-Morales et al., “Functional assessment of current upper limb prostheses: An integrated clinical and technological perspective,” *PLOS One*, vol. 18, no. 8, p. e0289978, 2023, doi: 10.1371/journal.pone.0289978.
- [116] A. M. Simon et al., “User performance with a transradial multi-articulating hand prosthesis during pattern recognition and direct control home use,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 31, pp. 271–81, 2023, doi: 10.1109/TNSRE.2022.3221558.
- [117] I. Vujaklija et al., “Online mapping of EMG signals into kinematics by autoencoding,” *J Neuroeng Rehabil*, vol. 15, p. 21, 2018, doi: 10.1186/s12984-018-0363-1.
- [118] N. Jiang et al., “Is accurate mapping of EMG signals on kinematics needed for precise online myoelectric control?,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 3, pp. 549–58, 2014, doi: 10.1109/TNSRE.2013.2287383.
- [119] N. Jiang et al., “Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 3, pp. 501–10, 2014, doi: 10.1109/TNSRE.2013.2278411.

- [120] A. Ameri et al., “Real-time, simultaneous myoelectric control using a convolutional neural network,” *PLoS One*, vol. 13, no. 9, p. e0203835, 2018, doi: 10.1371/journal.pone.0203835.
- [121] P. Lukyanenko et al., “Stable, simultaneous and proportional 4-DoF prosthetic hand control via synergy-inspired linear interpolation: A case series,” *J Neuroeng Rehabil*, vol. 18, p. 50, 2021, doi: 10.1186/s12984-021-00833-3.
- [122] C. Piazza et al., “Evaluation of a simultaneous myoelectric control strategy for a multi-DoF transradial prosthesis,” *IEEE Trans Neural Syst Rehabil Eng*, 2020, doi: 10.1109/tnsre.2020.3016909.
- [123] Y. Teh and L. J. Hargrove, “Understanding limb position and external load effects on real-time pattern recognition control in amputees,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 28, no. 7, pp. 1605–13, 2020, doi: 10.1109/TNSRE.2020.2991643.
- [124] A. J. Young et al., “A comparison of the real-time controllability of pattern recognition to conventional myoelectric control for discrete and simultaneous movements,” *J Neuroeng Rehabil*, vol. 11, p. 5, 2014, doi: 10.1186/1743-0003-11-5.
- [125] A. M. Simon et al., “Target achievement control test: Evaluating real-time myoelectric pattern-recognition control of multifunctional upper-limb prostheses,” *J Rehabil Res Dev*, vol. 48, no. 6, pp. 619–27, 2011, doi: 10.1682/jrrd.2010.08.0149.
- [126] S. M. Wurth and L. J. Hargrove, “A real-time comparison between direct control, sequential pattern recognition control and simultaneous pattern recognition control using a fitts’ law style assessment procedure,” *J Neuroeng Rehabil*, vol. 11, p. 91, 2014, doi: 10.1186/1743-0003-11-91.
- [127] A. L. Fougner et al., “System training and assessment in simultaneous proportional myoelectric prosthesis control,” *J Neuroeng Rehabil*, vol. 11, p. 75, 2014, doi: 10.1186/1743-0003-11-75.

- [128] S. Dosen et al., “Cognitive vision system for control of dexterous prosthetic hands: Experimental evaluation,” *J Neuroeng Rehabil*, vol. 7, p. 42, 2010, doi: 10.1186/1743-0003-7-42.
- [129] D. Johansen et al., “Control of a robotic hand using a tongue control system-A prosthesis application,” *IEEE Trans Biomed Eng*, vol. 63, no. 7, pp. 1368–76, 2016, doi: 10.1109/TBME.2016.2517742.
- [130] S. Amsuess et al., “Context-dependent upper limb prosthesis control for natural and robust use,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 24, no. 7, pp. 744–53, 2016, doi: 10.1109/TNSRE.2015.2454240.
- [131] S. Amsuess et al., “A multi-class proportional myocontrol algorithm for upper limb prosthesis control: Validation in real-life scenarios on amputees,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 23, no. 5, pp. 827–36, 2015, doi: 10.1109/TNSRE.2014.2361478.
- [132] C. Cipriani et al., “Closed-loop controller for a bio-inspired multi-fingered underactuated prosthesis,” *Proc IEEE Int Conf Robot Autom (ICRA)*, pp. 2111–6, 2006, doi: 10.1109/ROBOT.2006.1642016.
- [133] S. Amsuess et al., “Extending mode switching to multiple degrees of freedom in hand prosthesis control is not efficient,” *Proc 36<sup>th</sup> Annu Int Conf IEEE Eng Med Biol Soc (EMBC)*, pp. 658–61, 2014, doi: 10.1109/EMBC.2014.6943677.
- [134] M. G. Catalano et al., “Adaptive synergies for the design and control of the Pisa/IIT SoftHand,” *Int J Robot Res*, vol. 33, no. 5, pp. 768–82, 2014, doi: 10.1177/0278364913518998.
- [135] G. Matrone et al., “Two-channel real-time EMG control of a dexterous hand prosthesis,” *Proc 5<sup>th</sup> Int IEEE EMBS Conf Neural Eng (NER)*, pp. 554–7, 2011, doi: 10.1109/NER.2011.5910608.
- [136] P. D. Marasco et al., “Neurobotic fusion of prosthetic touch, kinesthesia, and movement promotes intrinsic,” *Sci Robot*, vol. 6, p. eabf3368, 2021, doi: 10.1126/scirobotics.abf3368.

- [137] A. M. Valevicius et al., “Characterization of normative angular joint kinematics during two functional upper limb tasks,” *Gait Posture*, vol. 69, pp. 176–86, 2019, doi: 10.1016/j.gaitpost.2019.01.037.
- [138] E. B. Lavoie et al., “Using synchronized eye and motion tracking to determine high-precision eye-movement patterns during object-interaction tasks,” *J Vis*, vol. 18, no. 6, p. 18, 2018, doi: 10.1167/18.6.18.
- [139] J. S. Hebert et al., “Quantitative eye gaze and movement differences in visuomotor adaptations to varying task demands among upper-extremity prosthesis users,” *JAMA Netw Open*, vol. 2, no. 9, p. e1911197, 2019, doi: 10.1001/jamanetworkopen.2019.11197.
- [140] A. M. Valevicius et al., “Compensatory strategies of body-powered prosthesis users reveal primary reliance on trunk motion and relation to skill level,” *Clin Biomech*, vol. 72, pp. 122–9, 2020, doi: 10.1016/j.clinbiomech.2019.12.002.
- [141] G. K. Patel et al., “Context-dependent adaptation improves robustness of myoelectric control for upper-limb prostheses,” *J Neural Eng*, vol. 14, no. 5, p. 056016, 2017, doi: 10.1088/1741-2552/aa7e82.
- [142] Q. Fu and M. Santello, “Improving fine control of grasping force during hand-object interactions for a soft synergy-inspired myoelectric prosthetic hand,” *Front Neurobot*, vol. 11, p. 71, 2018, doi: 10.3389/fnbot.2017.00071.
- [143] W. Zhang et al., “Cognitive workload in conventional direct control vs. pattern recognition control of an upper-limb prosthesis,” *Proc IEEE Int Conf Syst Man Cybern (SMC)*, pp. 2335–40, 2017, doi: 10.1109/SMC.2016.7844587.
- [144] K. A. Ingraham et al., “Evaluating physiological signal salience for estimating metabolic energy cost from wearable sensors,” *J Appl Physiol*, vol. 126, no. 3, pp. 717–29, 2018, doi: 10.1152/jappphysiol.00714.2018.
- [145] S. G. Hart, “NASA-task load index (NASA-TLX); 20 years later,” *Proc Hum Factors Ergon Soc Annu Meet (HFES)*, pp. 904–8, 2006, doi: 10.1177/154193120605000909.

- [146] J. Park and M. Zahabi, "Cognitive workload assessment of prosthetic devices: A review of literature and meta-analysis," *IEEE Trans Hum Mach Syst*, vol. 52, no. 2, pp. 181–95, 2022, doi: 10.1109/THMS.2022.3143998.
- [147] D. J. A. Brenneis et al., "The effect of an automatically levelling wrist control system," *Proc IEEE 16<sup>th</sup> Int Conf Rehabil Robot (ICORR)*, pp. 816–23, 2019, doi: 10.1109/ICORR.2019.8779444.
- [148] J. S. Hebert and J. Lewicke, "Case report of modified box and blocks test with motion capture to measure prosthetic function," *J Rehabil Res Dev*, vol. 49, no. 8, pp. 1163–74, 2012, doi: 10.1682/JRRD.2011.10.0207.
- [149] J. S. Hebert et al., "Normative data for modified box and blocks test measuring upper-limb function via motion capture," *J Rehabil Res Dev*, vol. 51, no. 6, pp. 918–32, 2014, doi: 10.1682/JRRD.2013.10.0228.
- [150] Q. A. Boser et al., "Cluster-based upper body marker models for three-dimensional kinematic analysis: Comparison with an anatomical model and reliability analysis," *J Biomech*, vol. 72, pp. 228–34, 2018, doi: 10.1016/j.jbiomech.2018.02.028.
- [151] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," *Adv Psychol*, vol. 52, pp. 139–83, 1988, doi: 10.1016/S0166-4115(08)62386-9.
- [152] J. V. V. Parr et al., "A tool for measuring mental workload during prosthesis use: The prosthesis task load index (PROS-TLX)," *PLOS One*, vol. 18, no. 5, p. e0285382, 2023, doi: 10.1371/journal.pone.0285382.
- [153] S. L. Wang et al., "Comparison of motion analysis systems in tracking upper body movement of myoelectric bypass prosthesis users," *Sensors (Switzerland)*, vol. 22, no. 8, p. 2953, 2022, doi: 10.3390/s22082953.
- [154] H. E. Williams et al., "Myoelectric prosthesis control using recurrent convolutional neural network regression mitigates the limb position effect," *bioRxiv*, 2024, doi: 10.1101/2024.02.05.578477.

- [155] J. Cowley et al., “Movement quality of conventional prostheses and the DEKA arm during everyday tasks,” *Prosthet Orthot Int*, vol. 41, no. 1, pp. 33–40, 2017, doi: 10.1177/0309364616631348.
- [156] E. Biddiss et al., “Consumer design priorities for upper limb prosthetics,” *Disabil Rehabil Assist Technol*, vol. 2, no. 6, pp. 346–57, 2007, doi: 10.1080/17483100701714733.
- [157] C. H. Jang et al., “A survey on activities of daily living and occupations of upper extremity amputees,” *Ann Rehabil Med*, vol. 35, no. 6, pp. 907–21, 2011, doi: 10.5535/arm.2011.35.6.907.
- [158] Z. Chen et al., “A review of myoelectric control for prosthetic hand manipulation,” *Biomimetics*, vol. 8, no. 3, p. 328, 2023, doi: 10.3390/biomimetics8030328.
- [159] A. A. Adewuyi et al., “Resolving the effect of wrist position on myoelectric pattern recognition control,” *J Neuroeng Rehabil*, vol. 14, p. 39, 2017, doi: 10.1186/s12984-017-0246-x.
- [160] M. D. Paskett et al., “Activities of daily living with bionic arm improved by combination training and latching filter in prosthesis control comparison,” *J Neuroeng Rehabil*, vol. 18, p. 45, 2021, doi: 10.1186/s12984-021-00839-x.
- [161] H. E. Williams et al., “Evidence that a deep learning regression-based controller mitigates the limb position effect for an individual with transradial amputation,” accepted to *Proc Myoelect Controls Symp (MEC24)*, 2024.
- [162] K. Ziegler-Graham et al., “Estimating the prevalence of limb loss in the United States: 2005 to 2050,” *Arch Phys Med Rehabil*, vol. 89, no. 3, pp. 422–9, 2008, doi: 10.1016/j.apmr.2007.11.005.

# Appendix A.

## Offline Analysis Prediction Accuracies

This appendix contains the mean limb position classification accuracies (Table A-1), mean movement classification accuracies (Table A-2), and mean movement regression  $R^2$  values (Table A-3) resulting from this study.

Table A-1. Mean limb position classification accuracies (across participants) using RCNN classification and LDA classification. Accuracies are provided for each combination of data streams: all EMG and IMU data streams from both Myo armbands (All Data Streams); all EMG data streams from both Myo armbands (EMG); all IMU data streams from both Myo armbands (IMU); and only accelerometer data streams from both Myo armbands (Accelerometer).

<b>Data Streams</b>	<b>RCNN Accuracy (%)</b>	<b>LDA Accuracy (%)</b>
All Data Streams	97.48	86.07
EMG	85.83	79.59
IMU	97.60	88.04
Accelerometer	99.01	98.66

Table A-2. Mean movement classification accuracies (across participants) using RCNN classification and LDA classification, under each specification: the baseline classifier (BC), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Accuracies are provided for each combination of data streams: data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands (when applicable).

<b>Classifier Specification</b>	<b>Data Streams</b>	<b>RCNN Accuracy (%)</b>	<b>LDA Accuracy (%)</b>
BC	Forearm EMG	85.64	85.14
	Both EMG	76.52	77.34
S1	Forearm EMG	94.11	93.09
	Both EMG	95.76	93.86
S2	Forearm EMG + Forearm Accelerometer	98.46	97.30
	Both EMG + Both Accelerometer	97.99	97.53
	Forearm EMG + Both Accelerometer	99.00	97.67
S3	Forearm EMG + Forearm Accelerometer	95.09	95.13
	Both EMG + Both Accelerometer	95.33	94.77
	Forearm EMG + Both Accelerometer	95.53	94.93

Table A-3. Mean movement regression  $R^2$  values (across participants) using the RCNN flexion/extension regression, SVR flexion/extension regression, RCNN pronation/supination regression, and SVR pronation/supination regression, under each specification: the baseline regressor (BR), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Accuracies are provided for each combination of data streams: data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands (when applicable).

<b>Regressor Specification</b>	<b>Data Streams</b>	<b>Flexion/Extension <math>R^2</math> (%)</b>		<b>Pronation/Supination <math>R^2</math> (%)</b>	
		<b>RCNN</b>	<b>SVR</b>	<b>RCNN</b>	<b>SVR</b>
BR	Forearm EMG	77.85	76.54	38.98	20.28
	Both EMG	71.84	77.02	36.89	25.30
S1	Forearm EMG	84.73	76.18	63.77	21.95
	Both EMG	85.58	77.16	69.13	35.08
S2	Forearm EMG + Forearm Accelerometer	84.93	77.26	84.97	60.73
	Both EMG + Both Accelerometer	84.04	77.74	86.27	65.11
	Forearm EMG + Both Accelerometer	84.73	77.45	85.72	62.41
S3	Forearm EMG + Forearm Accelerometer	84.34	78.43	59.32	33.41
	Both EMG + Both Accelerometer	84.78	79.34	66.65	47.51
	Forearm EMG + Both Accelerometer	84.39	78.58	61.40	34.72

# Appendix B.

## Offline Analysis Training and Prediction Times

This appendix contains the mean limb position classification training and prediction times (Table B-1), movement classification training and prediction times (Table B-2), and movement regression training and prediction times (Table B-3) resulting from this study.

Table B-1. Mean limb position classification training and prediction times (across participants) using RCNN classification and LDA classification. Times are provided for each combination of data streams: all EMG and IMU data streams from both Myo armbands (All Data Streams); all EMG data streams from both Myo armbands (EMG); all IMU data streams from both Myo armbands (IMU); and only accelerometer data streams from both Myo armbands (Accelerometer).

<b>Data Streams</b>	<b>RCNN Training Time (min)</b>	<b>RCNN Prediction Time (ms)</b>	<b>LDA Training Time (ms)</b>	<b>LDA Prediction Time (ms)</b>
All Data Streams	2.52	0.12	89.19	0.02
EMG	1.40	0.11	76.57	0.01
IMU	1.54	0.11	48.73	0.01
Accelerometer	1.68	0.10	38.48	0.01

Table B-2. Mean movement classification training and prediction times (across participants) using RCNN classification and LDA classification, under each specification: the baseline classifier (BC), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Times are provided for each combination of data streams: data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands (when applicable).

<b>Classifier Specification</b>	<b>Data Streams</b>	<b>RCNN Training Time (min)</b>	<b>RCNN Prediction Time (ms)</b>	<b>LDA Training Time (ms)</b>	<b>LDA Prediction Time (ms)</b>
BC	Forearm EMG	1.97	0.10	18.61	0.002
	Both EMG	1.67	0.11	20.15	0.003
S1	Forearm EMG	2.02	0.10	23.08	0.002
	Both EMG	1.88	0.11	28.41	0.003
S2	Forearm EMG + Forearm Accelerometer	1.66	0.10	24.03	0.002
	Both EMG + Both Accelerometer	1.98	0.11	30.28	0.004
	Forearm EMG + Both Accelerometer	1.95	0.11	24.54	0.002
S3	Forearm EMG + Forearm Accelerometer	9.07	5.24	81.29	2.07
	Both EMG + Both Accelerometer	8.98	5.29	88.52	2.18
	Forearm EMG + Both Accelerometer	9.79	5.26	81.53	2.08

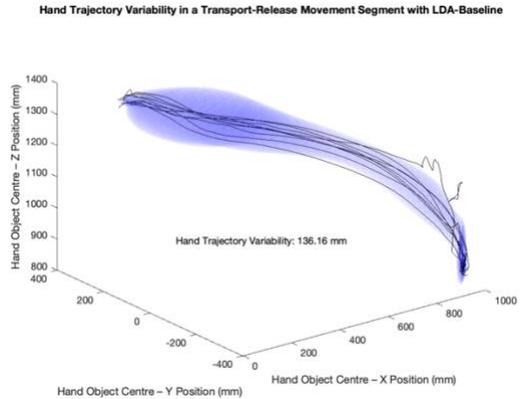
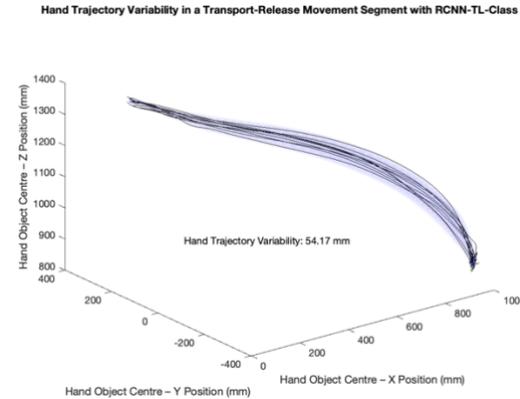
Table B-3. Mean movement regression training and prediction times (across participants) using RCNN regression and SVR regression, under each specification: the baseline regressor (BR), specification 1 (S1), specification 2 (S2), and specification 3 (S3). Times are provided for each combination of data streams: data from only the forearm Myo armband; data from both Myo armbands; and EMG data from the forearm and accelerometer data from both Myo armbands (when applicable).

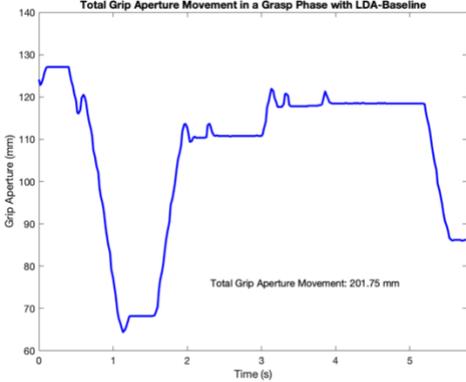
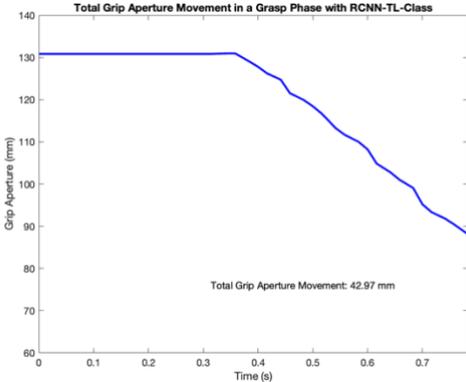
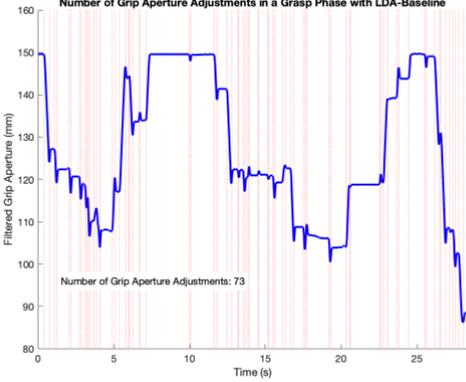
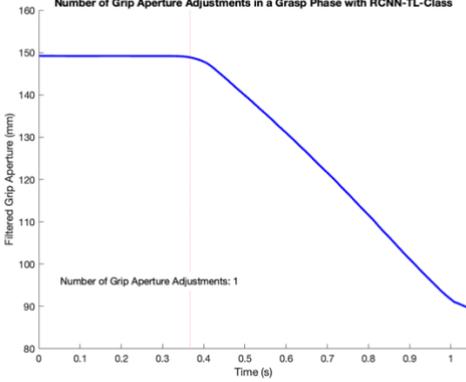
<b>Regressor Specification</b>	<b>Data Streams</b>	<b>RCNN Training Time (min)</b>	<b>RCNN Prediction Time (ms)</b>	<b>SVR Training Time (s)</b>	<b>SVR Prediction Time (ms)</b>
BR	Forearm EMG	0.48	4.43	2.41	0.56
	Both EMG	0.38	4.45	1.76	0.77
S1	Forearm EMG	0.94	4.42	26.38	1.32
	Both EMG	0.86	4.45	16.88	2.07
S2	Forearm EMG + Forearm Accelerometer	0.69	4.43	21.59	1.36
	Both EMG + Both Accelerometer	0.76	4.47	18.15	2.22
	Forearm EMG + Both Accelerometer	0.77	4.44	19.63	1.37
S3	Forearm EMG + Forearm Accelerometer	2.49	5.04	9.24	1.70
	Both EMG + Both Accelerometer	2.87	5.06	6.50	1.92
	Forearm EMG + Both Accelerometer	3.19	5.03	9.24	1.71

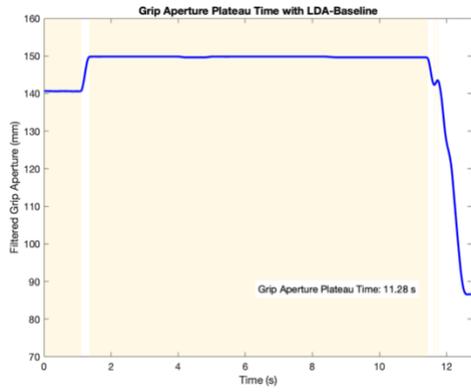
# Appendix C.

## Table of Metrics Examples

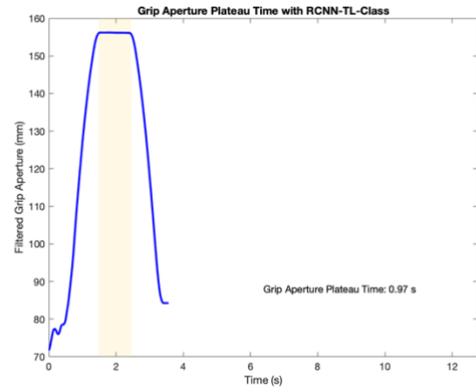
This table contains figures that exemplify “good” and “poor” results for selected metrics. For each metric presented, a description of the example figure precedes that associated good/poor graphs.

	Poor Result Example	Good Result Example
Hand Trajectory Variability	<p>The black lines illustrate a given participant's hand movement paths throughout the movement 3 Reach-Grasp movement segments of Pasta in each trial. The blue shaded area illustrates the standard deviations at each sampled point. Both figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. The hand trajectory variability calculated in these examples are indicated in each figure.</p>  <p>Hand Trajectory Variability in a Transport-Release Movement Segment with LDA-Baseline</p> <p>Hand Trajectory Variability: 136.16 mm</p> <p>The poor result example illustrates a large standard deviation, resulting in a large hand trajectory variability (136.16 mm).</p>	 <p>Hand Trajectory Variability in a Transport-Release Movement Segment with RCNN-TL-Class</p> <p>Hand Trajectory Variability: 54.17 mm</p> <p>The good result example illustrates a smaller standard deviation, resulting in a smaller hand trajectory variability (54.17 mm).</p>
Total Grip	<p>The blue line illustrates the grip aperture throughout a movement 3 Grasp phase of Pasta. Both figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. The total grip aperture movement are presented in each figure.</p>	

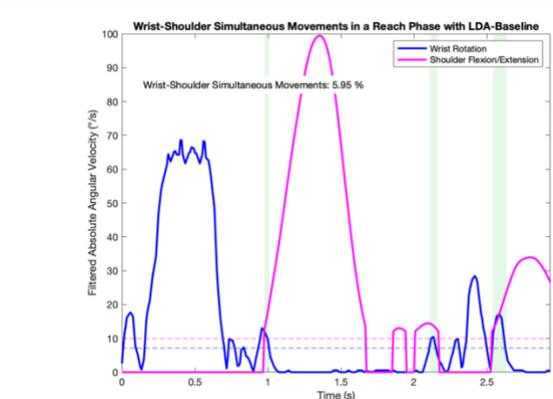
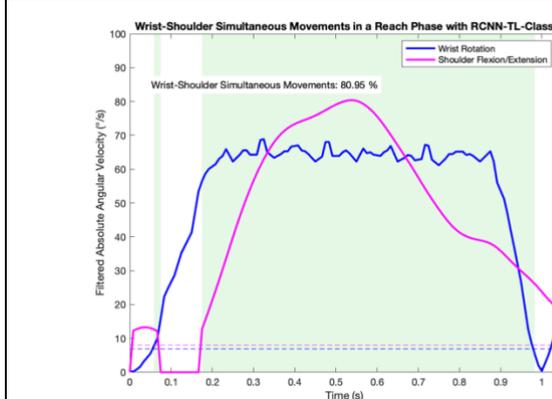
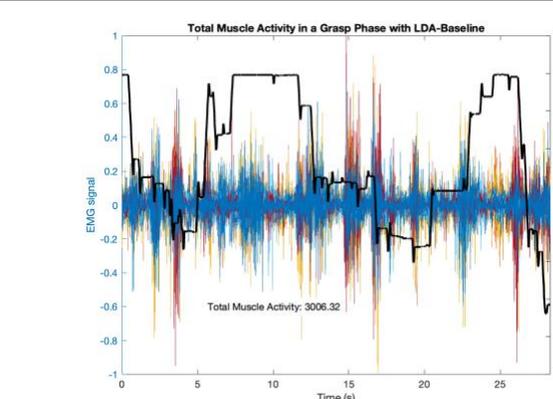
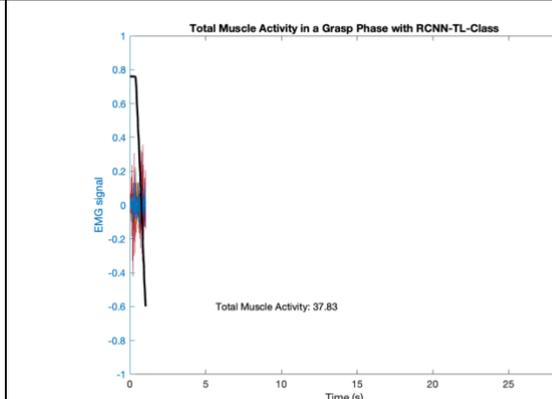
	 <p>The poor result example illustrates hand openings throughout the grasp phase, resulting in a large total grip aperture movement (201.75 mm).</p>	 <p>The good result example illustrates a grip aperture movement with less fluctuation. This resulted in a smaller total grip aperture movement (42.97 mm).</p>
	<b>Poor Result Example</b>	<b>Good Result Example</b>
<b>Number of Grip Aperture Adjustments</b>	<p>The blue line illustrates the filtered grip aperture throughout a movement 3 Grasp phase of Pasta. The dashed vertical lines indicate grip aperture adjustments that were detected. Both figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. The number of grip aperture adjustments are presented in each figure.</p>	
	 <p>The poor result example identifies several adjustments (73).</p>	 <p>The good result example identifies only 1 adjustment, at which the hand first started to close.</p>
<b>Grip Aperture</b>	<p>The blue line illustrates the grip aperture throughout a movement 3 Reach-Grasp movement segment of Pasta. The shaded area indicates the portions of the movement segment in which the grip aperture plateau requirements were met. Both figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. Note that the same x axis scale was used in both figures to further illustrate the grip aperture plateau. The grip aperture plateau time is presented in each figure.</p>	



The poor result example identifies a long grip aperture plateau time (11.28 s).



The good result example identifies a short grip aperture plateau time (0.97 s).

	Poor Result Example	Good Result Example
Wrist-Shoulder Simultaneous Movements	<p>The solid blue line illustrates the filtered absolute wrist rotation angular velocity throughout a movement 3 Reach phase of RCRT down, and the solid magenta line illustrates the filtered absolute shoulder flexion/extension angular velocity throughout that same phase. The dashed horizontal blue and magenta lines indicate the thresholds corresponding to these angular velocities. Finally, the shaded areas indicate the portions of the phase in which both angular velocities were above their respective thresholds. The figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. The wrist-shoulder simultaneous movements are indicated in each figure.</p>  <p>The poor result example identifies wrist rotation control when the shoulder is usually not moving, resulting in a small wrist-shoulder simultaneous movement (5.95%).</p>	 <p>The good result example identifies wrist rotation control during shoulder movements, resulting in a large wrist-shoulder simultaneous movement (80.95%).</p>
Total Muscle Activity	<p>The 8 coloured lines illustrate the 8 EMG signals throughout a movement 3 Grasp phase of Pasta. The black line illustrates the Grip Aperture throughout that same phase, to aid in the understanding of the EMG signals. Both figures were generated using data from the same participant, with the poor result example illustrating a trial in which they were using the LDA-Baseline and the good result example illustrating a trial in which they were using the RCNN-TL. Note that the same x axis scale was used in both figures to further illustrate the differences in time required to close the hand. The total muscle activity is indicated in each figure.</p>  <p>The poor result example identifies difficulties in closing the hand, resulting in a long phase and strong EMG signals. So, a large total muscle activity resulted (3006.32).</p>	 <p>The good result example identifies more ease in closing the hand, resulting in a shorter phase and weaker EMG signals. So, a smaller total muscle activity resulted (37.83).</p>

## Appendix D.

# Detailed Controller Comparison Results

This appendix outlines results for both controller comparisons, all three tasks, and in all three groups of metrics (Task Performance, Control Characteristics, and User Experience). The two controller comparisons are as follows:

- The recurrent convolutional neural network classification controller (RCNN-Class) versus the linear discriminant analysis baseline classification control (LDA-Baseline)
- The recurrent convolutional neural network regression controller (RCNN-Reg) versus LDA-Baseline

The controller comparison results are detailed in the three tables below—one table for each group of metrics. Each table displays the median within-participant differences between the controllers in question, calculated as RCNN-Class minus LDA-Baseline or RCNN-Reg minus LDA-Baseline. Interquartile ranges of these differences are presented in parentheses. Green cells indicate metrics in which RCNN-Class or RCNN-Reg performed significantly better than LDA-Baseline ( $p < 0.05$ ). Grey cells indicate instances where a metric was not relevant. Dark cell borders indicate metrics that displayed evidence of the limb position effect under LDA-Baseline control.

Table D-1. Task Performance Results

Controller Comparison		RCNN-Class versus LDA-Baseline			RCNN-Reg versus LDA-Baseline		
Task		Pasta	RCRT Up	RCRT Down	Pasta	RCRT Up	RCRT Down
Success Rate		0.00 (15.00)	5.00 (10.00)	5.00 (25.00)	0.00 (17.50)	0.00 (17.50)	0.00 (7.50)
Trial Duration (s)		-0.93 (10.60)	1.04 (10.18)	-8.78 (14.40)	-3.03 (11.73)	-2.13 (7.35)	-5.43 (16.64)
Phase Duration (s)	Reach	0.11 (0.76)	-0.03 (1.23)	0.07 (0.74)	-0.39 (0.75)	-0.63 (0.90)	-0.14 (1.07)
	Grasp	-0.14 (1.01)	-0.12 (1.35)	-0.44 (1.35)	-0.27 (0.98)	-0.37 (1.02)	-0.22 (1.61)
	Transport	-0.03 (0.40)	0.06 (1.25)	-0.10 (1.40)	-0.34 (0.78)	0.28 (1.23)	-0.65 (1.43)
	Release	-0.06 (0.41)	0.20 (1.16)	-0.29 (2.47)	-0.09 (1.03)	0.05 (1.55)	-0.10 (1.07)
Relative Phase Duration (%)	Reach	0.99 (6.25)	-0.72 (7.30)	3.79 (12.01)	-1.07 (6.39)	-6.51 (5.46)	3.11 (9.08)
	Grasp	-2.35 (8.65)	-1.75 (7.68)	-0.37 (12.47)	-2.77 (7.66)	1.15 (13.75)	0.99 (8.51)
	Transport	1.03 (7.17)	-0.47 (8.35)	2.07 (6.47)	3.20 (6.39)	4.15 (5.45)	-1.02 (9.44)
	Release	0.91 (4.95)	0.28 (5.32)	-1.73 (12.30)	0.82 (8.56)	2.27 (8.17)	0.16 (8.38)
Peak Hand Velocity (mm/s)	Reach-Grasp	32.71 (163.96)	-21.21 (128.75)	-35.18 (252.92)	99.01 (335.58)	111.25 (152.43)	49.38 (216.65)
	Transport-Release	-20.45 (110.08)	-36.87 (110.66)	-39.07 (129.35)	117.66 (269.54)	4.13 (141.52)	46.48 (102.92)
Hand Distance Travelled (mm)	Reach-Grasp	-22.01 (180.61)	0.26 (84.33)	-16.69 (63.15)	-30.85 (64.24)	-27.17 (50.41)	-6.40 (98.83)
	Transport-Release	-45.52 (70.92)	10.79 (116.79)	-31.29 (151.25)	-9.99 (41.14)	-1.14 (52.19)	-39.99 (48.94)
Hand Trajectory Variability (mm)	Reach-Grasp	-7.85 (43.10)	-3.02 (10.15)	-14.04 (30.23)	-21.77 (30.55)	-7.48 (28.83)	-8.01 (25.14)
	Transport-Release	-17.11 (53.22)	1.98 (22.24)	-2.54 (21.07)	-6.06 (47.51)	-2.96 (16.29)	-8.54 (15.74)

Table D-2. Control Characteristics Results

Controller Comparison		RCNN-Class versus LDA-Baseline			RCNN-Reg versus LDA-Baseline		
Task		Pasta	RCRT Up	RCRT Down	Pasta	RCRT Up	RCRT Down
Total Grip Aperture Movement (mm)	Reach	7.21 (13.11)	-3.53 (14.19)	6.34 (26.61)	0.72 (20.14)	-8.07 (24.73)	-0.41 (38.98)
	Grasp	5.30 (13.85)	-1.63 (17.87)	-5.61 (22.50)	-4.33 (12.70)	-9.38 (25.33)	1.40 (22.87)
	Transport	1.45 (9.79)	2.06 (10.31)	0.82 (12.50)	-2.48 (11.95)	-2.47 (12.35)	-3.41 (6.84)
	Release	2.16 (11.57)	3.62 (14.30)	-2.13 (12.53)	-9.41 (31.62)	-14.30 (12.79)	-21.29 (8.90)
Total Wrist Rotation Movement (deg)	Reach	-2.81 (50.23)	2.28 (47.44)	-2.36 (35.41)	-1.19 (25.64)	-30.25 (31.33)	9.48 (32.87)
	Grasp	-10.22 (37.46)	-6.53 (32.63)	-2.17 (44.77)	-20.21 (37.39)	2.05 (40.14)	-0.37 (71.02)
	Transport	-5.98 (43.54)	7.49 (41.55)	-8.63 (31.44)	-1.90 (34.57)	-2.78 (30.73)	-42.21 (59.37)
	Release	2.97 (25.76)	6.67 (36.36)	-7.76 (59.42)	-5.45 (25.19)	-11.32 (18.09)	2.40 (70.81)
Number of Grip Aperture Adjustments	Reach	0.42 (1.03)	0.04 (1.22)	0.06 (2.00)	0.30 (2.16)	-0.38 (2.07)	0.10 (1.66)
	Grasp	-0.60 (3.03)	-0.21 (3.81)	-0.58 (2.41)	-1.64 (2.26)	-2.21 (2.12)	-1.50 (3.61)
	Transport	0.31 (1.87)	0.74 (2.53)	0.40 (1.91)	-0.61 (1.54)	-0.10 (5.03)	-1.11 (1.37)
	Release	0.28 (1.72)	0.27 (1.63)	0.11 (2.76)	-1.50 (1.98)	-1.00 (2.50)	-0.70 (1.96)
Number of Wrist Rotation Adjustments	Reach	0.19 (2.79)	-0.69 (3.53)	-0.63 (3.34)	-0.20 (2.81)	-2.12 (2.25)	-0.89 (2.37)
	Grasp	-2.19 (5.51)	-0.49 (6.35)	-1.02 (4.85)	-3.89 (4.82)	-3.50 (3.56)	-3.33 (7.74)
	Transport	0.32 (2.71)	-0.10 (3.46)	-1.04 (5.52)	-0.90 (5.88)	-3.60 (2.66)	-4.72 (3.43)
	Release	0.78 (3.66)	1.34 (4.25)	-1.07 (7.70)	-1.59 (2.48)	-3.00 (2.71)	-4.37 (3.00)
Grip Aperture Plateaus (s)	Reach-Grasp	-0.15 (1.41)	-0.13 (1.20)	-0.27 (1.27)	-0.51 (1.15)	-0.64 (0.88)	-0.69 (1.32)
Wrist-Shoulder Simultaneous Movements (%)	Reach		0.04 (0.17)	-0.03 (0.15)		0.03 (0.42)	0.30 (0.23)
	Transport		-0.03 (0.15)	-0.01 (0.11)		0.11 (0.21)	0.07 (0.22)
Total Muscle Activity	Reach	5.04 (129.59)	-21.95 (170.04)	5.31 (123.63)	-38.75 (104.36)	-86.99 (159.03)	-13.78 (181.33)
	Grasp	-63.74 (118.36)	-56.77 (95.99)	-49.43 (174.33)	-35.67 (132.40)	-57.07 (208.64)	-21.62 (150.71)
	Transport	2.20 (83.41)	22.61 (125.16)	-49.64 (184.25)	-13.10 (127.34)	66.54 (241.71)	-39.36 (219.93)
	Release	-9.37 (73.74)	-1.41 (154.04)	-67.13 (342.45)	-34.38 (134.07)	-15.10 (187.42)	0.36 (185.66)

Table D-3. User Experience Results

Controller Comparison		<b>RCNN-Class versus LDA-Baseline</b>	<b>RCNN-Reg versus LDA-Baseline</b>
NASA-TLX	Mental Demand	-10.00 (22.50)	20.00 (22.50)
	Physical Demand	-10.00 (27.50)	5.00 (23.75)
	Temporal Demand	5.00 (22.50)	-10.00 (41.25)
	Performance	-5.00 (12.50)	-5.00 (13.75)
	Effort	-2.50 (27.50)	0.00 (26.25)
	Frustration	-5.00 (40.00)	-5.00 (13.75)
Usability	Intuitiveness	0.40 (0.95)	0.00 (0.88)
	Effectiveness - RCRT	-0.25 (0.85)	0.62 (1.15)
	Effectiveness - Pasta	0.40 (1.25)	0.62 (1.17)
	Reliability	0.00 (1.40)	0.38 (2.02)