User-Friendly Robust MPC Tuning for Uncertain Paper-Making Processes

by

Ning He

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

Doctor of Philosophy

in

Control systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Paper-making systems are known to be large-scale and two-dimensional. In such systems, the control performance needs to be ensured at both the machine direction (MD) and cross direction (CD) for successful paper production. This thesis studies the robust tuning problem of MD and CD Model Predictive Control (MPC) systems, and focuses on paper-making processes with uncertain model parameters.

Four problems are solved in the work. First, the robust MD-MPC tuning for uncertain single-input, single-output (SISO) paper-making processes is explored. A sufficient condition is first derived for robust stability of MD-MPC, and then an automatic tuning procedure is proposed to achieve satisfactory closed-loop responses, as measured by overshoots, settling times and output oscillations with user-specified parametric uncertainties. Second, the robust MD-MPC tuning for uncertain multiple-input, multiple-output (MIMO) paper-making processes is studied. An efficient visualization technique is developed to characterize the set of all possible step responses for all outputs given the parametric uncertainty. An automatic tuning algorithm is then developed to achieve the desired time domain performance. In addition, a technique to predict the computation time of the tuning algorithm is proposed. Third, the robust spatial tuning of CD-MPC is investigated. The weighting matrix S_b is first appropriately designed to suppress high frequency components in the actuator profile. A systematic tuning procedure is then developed to adjust the corresponding multipliers to guarantee the robust stability and to reduce the variability of the measurement profile given the pre-specified parametric uncertainties. Lastly, the robust temporal tuning of CD-MPC is studied. A performance visualization technique is proposed to evaluate all possible 2σ (two times of the standard deviation) spreads of the measurement and actuator profiles given the pre-specified parametric uncertainties. Then, a temporal filter is adopted to smooth the MPC reference trajectory, and a systematic procedure is developed to tune the parameter in the temporal filter for robust stability and satisfactory 2σ performance.

The effectiveness of the proposed tuning algorithms is verified through industrial examples extracted from the pulp and paper industry. By utilizing the proposed techniques, the MPC tuning parameters can be automatically determined to meet intuitive robust performance specifications for paper-making processes with easy-to-understand parametric uncertainties.

Preface

The research results presented in this thesis were based on a national research collaboration with Mr. Johan Backström and Dr. Michael Forbes from Honeywell Process Solutions, Vancouver. In addition, the research in Chapters 2-3 was also collaborated with Dr. Dawei Shi, and the research in Chapter 2 was further collaborated with Dr. Jiadong Wang, and the research in Chapters 4-5 was also collaborated with Dr. Xiaotao Liu, all from University of Alberta. The research idea in Chapter 2 was jointly discussed with Dr. Jiadong Wang and Dr. Dawei Shi. The research in Chapters 3-5 was based on my original ideas. Mr. Johan Backström, Dr. Michael Forbes and Dr. Xiaotao Liu had provided valuable suggestions for the research. The proposed results including the algorithms, derivations, analysis, verifications as well as the introduction and literature review in Chapter 1 were my original research work.

- The results in Chapter 2 have been published as He, N., Shi, D., Wang, J., Forbes, M., Backstrm, J., & Chen, T., "Automated two-degree-of-freedom MPC tuning," *Industrial & Engineering Chemistry Research*, 54(43), 10811-10824, 2015. A preliminary version has been published as He, N., Shi, D., Wang, J., Forbes, M., Backstrm, J., & Chen, T., "User friendly robust MPC tuning of uncertain paper-making processes," In *Proceedings of the 9th International Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 1022-1027, 2015.

- The results in Chapter 3 have been published as He, N., Shi, D., Forbes, M., Backstrm, J., & Chen, T., "Robust tuning for machine-directional predictive control of MIMO paper-making processes," *Control Engineering Practice*, 55, 1-12, 2016. The research idea in Chapter 3 has also been granted as a US patent as He, N., Shi, D., Forbes, M., Backstrm, J., & Chen, T., "Method and apparatus for robust tuning of model-based process controllers used with multiple-input, multiple-output (MIMO) processes," *US patent*, Publication No.WO/2016/191849.

- The research idea in Chapter 4 has been submitted as a US patent as He, N., Liu, X., Forbes, M., Backstrm, J., & Chen, T., "Method of designing model predictive control for cross-directional flat sheet manufacturing processes to guarantee spatial robustness and to prevent actuator picketing," *US patent*, Application No. 15/273,709, 2016.

- The research idea in Chapter 5 has been submitted as a US patent as He, N., Liu, X., Forbes, M., Backstrm, J., & Chen, T., "Method of designing model predictive control for cross-directional flat sheet manufacturing processes to guarantee temporal robust stability and performance," *US patent*, Application No. 15/273,705, 2016. In addition, the results in Chapters 4-5 have been submitted for publication as He, N., Liu, X., Forbes, M., Backstrm, J., & Chen, T., "Robust tuning of cross-directional model predictive control for paper-making processes," *IEEE transactions on Control System Technology*, 2016.

Acknowledgements

First of all, I would like to express my sincere appreciation to my supervisor, Dr. Tongwen Chen, for all the time and effort he gave to my PhD research. He provided me a valuable opportunity to work in a great university-industry collaborative research project (with Honeywell Vancouver as the industrial partner), and patiently supervised me in doing every single piece of the research work, from reviewing the literature to developing our own methodology, from writing a research proposal to preparing a formal workshop presentation. Without his careful guidance and enormous support, I can hardly experience such a rewarding and interesting PhD study.

Secondly, my sincere gratitude also goes to all the collaborators in the Honeywell research project, Dr. Jiadong Wang, Dr. Dawei Shi, and Dr. Xiaotao Liu from University of Alberta, and Mr. Johan Backström and Dr. Michael Forbes from Honeywell Process Solutions, Vancouver, for all their kind help and insightful suggestions on the results we developed in the project. I have really learned a lot from them and it was an invaluable experience to work with excellent researchers from both academia and industry.

I would also like to thank the supervisory committee members and examiners, Dr. Mahdi Tavakoli, Dr. Qing Zhao, Dr. Jinfeng Liu from University of Alberta and Dr. Hugh Liu from University of Toronto, for their insightful suggestions on my thesis.

I also wish to thank all the colleagues in the Advanced Control Group and the kind staffs in the department for their kind help throughout my graduate program.

Finally, I would like to thank my parents, for all their support and love

over the years. To them I dedicate the thesis.

Financial support from Natural Sciences and Engineering Research Council of Canada, Honeywell Process Solutions, and China Scholarship Council is also gratefully acknowledged.

Contents

List of Tables

List of Figures

List of Acronyms

MPC	Model Predictive Control
MD	Machine Direction
CD	Cross Direction
SISO	Single Input Single Output
MIMO	Multiple Input Multiple Output
2-DOF	2 Degree Of Freedom
FOPDT	First Order Plus Dead Time
TV	Total Variation
MISO	Multiple Input Single Output
RCM	Rectangle Circulant Matrix
DFT	Discrete Fourier Transform
FIR	Finite Impulsive Response
DR	Decay Ratio
QP	Quadratic Programming

Chapter 1

Introduction

1.1 Paper Machine and Paper-Making Process

Pulp and paper industry is a traditional Multi-Billion-Dollar business. In paper-making process, the paper machine is the key component, which can convert the semi-liquid mixture of wood cellulose fibers and water into varieties of paper products. In Figure 1.1, the standard configuration of a modern (Fourdrinier style) paper machine is shown, and we can briefly describe the operating process of the machine as follows: a slurry of fibers and water at $0.25 \sim 0.55\%$ consistency[1] is first inputted to the headbox of the machine, and then the consistency is being increased through the wet end section, the press section, and the dryer section, after which the consistency is increased to $91 \sim 95\%$ and the slurry has already become a dry paper sheet. Then, the dry paper sheet is further manipulated within the post drying operation section for brightness and thickness, and finally collected in a reel at the end of the machine.

In paper-making with modern high-speed paper machines, sheet properties must be continuously monitored and controlled to guarantee that the paper product quality specifications are satisfied along both the Machine Direction (MD) and Cross Direction (CD) [13, 65]. Machine direction indicates the direction towards which the paper moves on the machine, and cross direction is the direction perpendicular to machine direction [13]. Correspondingly, two control problems are involved in the paper making process: MD control

[1] Consistency indicates the proportion of the wood fibers in the slurry.

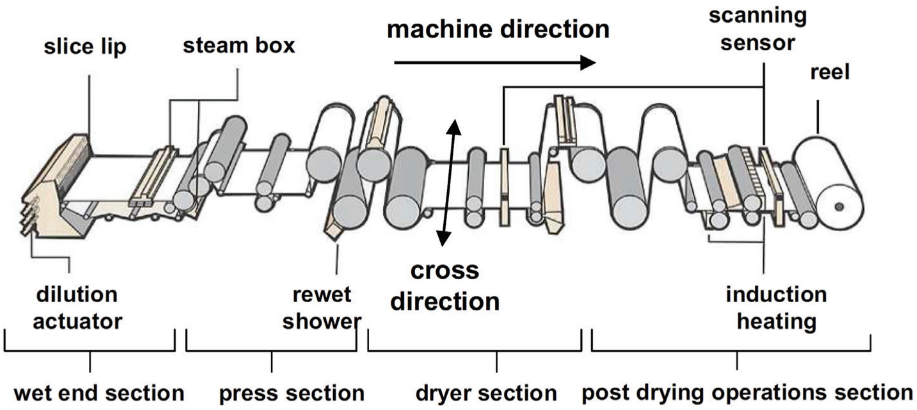


Figure 1.1: General view of a modern (Fourdrinier style) paper machine. (Artwork courtesy of Honeywell)

and CD control. MD control accounts for minimizing the variation of several physical properties of the paper product (e.g., dry weight and moisture) with a number of manipulated variables (e.g., stock flow, steam pressure and machine speed) in the machine direction. CD control solves a similar problem but with respect to the paper product qualities along cross direction. Various feedback control strategies are proposed for MD and CD control. Model predictive control (MPC), a control strategy which takes control constraints explicitly into consideration, has recently found its success in the pulp and paper industry [5, 6, 8]. MPC is an optimization-based control approach, and thus it involves a set of design parameters that are highly related to the robustness and the performance of the controlled system [45], making MPC tuning an important task. In this thesis, we focus on the MD and CD MPC tuning problems for paper-making processes with uncertain model parameters.

1.2 Literature Review

In this section, the existing MPC tuning methods are reviewed. As MD and CD processes have different system structures, the literature review for MD and CD MPC tuning is carried out separately.

1.2.1 Robust MPC tuning for MD processes

Both single input, single output (SISO) and multiple input, multiple output (MIMO) systems are included in the paper-making MD processes. The problem of MPC parameter tuning for SISO systems has been widely investigated in the literature. One type of the tuning approaches was to match the unconstrained MPC controller or the corresponding closed-loop transfer function with a desired controller or transfer function so that the desired closed-loop performance could be inherited, see, e.g., [10] and [48]. Another type of methods investigated the relationship between the closed-loop system responses and the effect of MPC tuning parameters based on the approximation of MPC solutions, see, e.g., [1] and [25]. In addition, some analytical results for MPC tuning were proposed in [7] and [64]. Among these approaches, only a small portion considered model uncertainty in MPC tuning. In [31], a min-max strategy was utilized to deal with model mismatch explicitly, based on which strong robustness could be obtained. In [39], by choosing the Morari resiliency index and the condition number as the performance measure, an efficient tuning approach was provided by utilizing the particle swarm optimization. In [59], the tuning parameters were calculated by finding an optimal bandwidth that provided a good trade-off between robustness and nominal performance. In [9], the authors proposed a modified generalized predictive control framework that was easy to implement and tune for the first-order-plus-dead-time (FOPDT) model, and the inherent robustness properties were also analyzed. In [58], the tuning problem was solved by balancing the contradictory control objectives in the optimization problem of the MPC. In [60], the tuning parameters for the nonlinear MPC were calculated based on muti-objective optimization and Pareto optimality. In [11], the authors proposed an MPC tuning strategy based on the frequency domain design, in which a min-max optimization approach was utilized to minimize the maximum singular value.

The parameter tuning problem becomes more difficult to deal with when MIMO plants and model mismatch are considered simultaneously [45]. In the existing literature, the generally utilized idea amounts to investigating the closed-loop performance with respect to each specific controller parameter based on the following two-step analysis: 1) hold all the parameters that affect the controller performance except one to make the tuning problem into a problem with one degree of freedom; 2) adjust this parameter to investigate the relationship between it and the closed-loop system behavior. Based on this type of investigation, some principles for multi-variable MPC tuning were developed, see [19], [25], also, [43], [47]. Besides, in [49], the authors proposed a systematic approach to adjust the MPC parameters by matching the closed-loop transfer function obtained via the unconstrained MPC with a desired transfer function. In [10], a systematic tuning method was developed, in which the controller parameters were tuned by minimizing the difference between the MIMO MPC controller and a pre-assigned multi-variable controller, and thus the features of the pre-assigned controller could be inherited. In [40], a metamorphic MPC framework was developed, in which a tuning parameter was incorporated such that one could move smoothly from an existing controller to a new MPC strategy. In addition, the readers can refer to [26] for a comprehensive review of the existing results on MPC tuning.

In the aforementioned robust tuning strategies, the performance indices and model uncertainty level are normally specified in the frequency domain, which are difficult to understand and specify by paper machine operators who may not have the background in robust control theory. Therefore, it is desirable that intuitive, time-domain performance specifications (e.g., overshoots and settling times in step responses) as well as easy-to-understand parametric uncertainties be directly incorporated into the tuning procedure to maintain the user-friendliness. Despite of the progress made in this area, such an easy-to-use robust MPC tuning approach is still missing and is desired in the pulp and paper industry.

1.2.2 Robust MPC tuning for CD processes

Different from MD processes that can be described by more standard models (a SISO MD process can be modeled as an FOPDT system while the MIMO one can be represented as the superposition of FOPDT components), CD processes are large scale two-dimensional (spatial and temporal) systems [20]. As suggested in process model identification algorithms [28, 30] and Theorem 3 in [20], the spatial response and temporal response are decoupled. Consequently, the controller tuning of CD processes can be separated into spatial tuning and temporal tuning. Spatial tuning aims to tune the weighting matrices such that the steady-state paper property across the paper sheet is consistent; temporal tuning concerns more about the satisfaction of time domain performance indices, e.g., settling times and overshoots, of the actuator and measurement profiles.

A lot of research results on spatial tuning have been reported. The spatial frequency concept of spatially-distributed CD processes was investigated thoroughly in [15], and the application of the spatial frequency was proposed in [41] for the controller design of CD processes. It was analyzed in [16] that the spatial frequency response of a single actuator determined the spatial frequency bandwidth of a CD process. A constructive procedure to design spatially-distributed feedback controllers was proposed in [57] and was applied in paper-making processes [56]. Model predictive control has been recently introduced into CD control in paper-making processes [5, 6, 8], with the advance of computational capability as well as the development of fast QP solvers [63]. Some stability margin and parameter tuning criteria were obtained via rectangular circulant matrices (RCMs) for the unconstrained CD-MPC, which provided a guide in the parameter tuning algorithms [20]. Furthermore, an approximate steady-state performance prediction technique was proposed to speed up the parameter tuning procedure [21] for the constrained CD-MPC. An automated tuning method was presented for the CD process such that the performance and robustness could be simultaneously satisfied under unstructured uncertainty [18]. An extensive review on CD control of sheet and film processes was given in [61]. It is worth noting that in the existing literature, no specific effort was dedicated to reducing the undesirable high frequency components that might exhibit in the actuator profile. As illustrated in [16, 46], the CD system has a limited spatial bandwidth. It is desirable that the limited spatial bandwidth property be explicitly incorporated into the parameter tuning procedure to avoid harmful high-frequency picketing in the control actuator beam.

Temporal tuning aims to tune the parameters such that the performance indices such as the settling time of the measurement profile and actuator profile overshoot, which can be easily understood by the technician-level operators and maintenance personnel, are satisfied. This can usually be realized by adding a weighted control effort in the associated cost function. However, if a smaller weighting parameter is chosen, then a larger actuator profile overshoot will be observed. On the other hand, if a larger weighting parameter is chosen, then it will take a longer time for the output profile to achieve a steady-state value. To the best of the author's knowledge, few research results have been published in the literature for the temporal tuning of CD processes with time domain performance indices.

Ubiquitously existing uncertainties make the spatial tuning and the temporal tuning more challenging. The parameters in CD processes are usually identified through bump tests [23, 27, 28] and are inevitably subject to parametric model uncertainties [29, 41]. Furthermore, geometric misalignments of CD actuators and measurement profiles, sheet wandering and paper shrinkage, etc, also give rise to some additional uncertainties to system models used in CD-MPC strategies. Thus, it is also necessary that spatial and temporal tuning algorithms be robust to the uncertainties mentioned above.

1.3 Contribution of the Thesis

In this thesis, the proposed techniques aim at providing solutions to the challenge of finding MPC tuning parameters to meet intuitive robust performance specifications for paper-making processes with easy-to-understand parametric uncertainties. Several user-friendly robust tuning algorithms are developed for MD and CD MPC, forming the main contributions of the thesis.

In Chapter 2, the automated tuning of a two-degree-of-freedom model predictive controller for SISO industrial MD processes with model uncertainties is explored. The objective of the tuning algorithm is to automatically determine the MPC tuning parameters such that 1) the robust stability can be guaranteed; 2) the worst-case overshoot is controlled; 3) the oscillations in process outputs are attenuated; and 4) the worst-case settling time is minimized. A rigorous robust stability analysis is first conducted based on the connection between parametric uncertainties and unstructured uncertainties, and a tight robust stability condition is derived. As the specification on process output variation is not easily made by the end users, two alternative methods are proposed to automatically determine the tolerable total variation, which lead to two automatic tuning algorithms that achieve the tuning objectives. The proposed results are tested and verified through examples extracted from industrial processes in the pulp and paper industry, and comparisons are made with other existing results.

In Chapter 3, we focus on the controller tuning problem of machine-directional predictive control for MIMO paper-making processes represented as superposition of FOPDT components with uncertain model parameters. A user-friendly multi-variable tuning problem is formulated based on user-specified time domain specifications and then simplified based on the structure of the closed-loop system. Based on the simplified tuning problem and a proposed performance evaluation technique, a fast multi-variable tuning technique is developed by ignoring the constraints of the MPC. In addition, a technique to predict the computation time of the tuning algorithm is proposed. The efficiency of the proposed method is verified through a Honeywell real time simulator platform with a MIMO paper-making process obtained from real data from an industrial site.

In Chapter 4, we focus on the robust spatial tuning of cross-directional model predictive control with model parameter uncertainties. The weighting matrix S_b in the MPC cost function is first appropriately designed to reduce the undesirable high frequency components that might exhibit in the actuator profile. A systematic tuning procedure is then developed to automatically determine the corresponding multipliers so that robust stability and reduced variability of the actuator and measurement profiles can be achieved given the pre-specified parametric uncertainties. The effectiveness of the proposed tuning algorithms is verified through the Honeywell real time simulator platform using a system model extracted from the pulp and paper industry.

In Chapter 5, the temporal tuning problem of cross-directional model predictive control under model-plant mismatch is studied. The user-friendly temporal performance indices are first defined based on the two times of the standard deviation (2σ) of the input/output profile and a visualization technique is proposed to evaluate all the possible 2σ performance given the pre-specified parametric uncertainty. Then, a temporal filter is adopted to smooth the MPC reference trajectory, and an automatic tuning algorithm is presented for the parameter in the temporal filter to guarantee satisfactory performance in terms of the proposed 2σ indices. In the presence of parametric uncertainties, the tuning is based on the worst-case situation, which ensures that the required performance indices can be satisfied. An application to a process extracted from the pulp and paper industry is employed to verify the effectiveness of the proposed algorithm.

Chapter 2

User-Friendly Robust Tuning for SISO MD-MPC[*]

2.1 Introduction

In this chapter, the automated tuning of a two-degree-of-freedom model predictive controller for single-input, single-output industrial MD processes with model uncertainties is explored. The target of MD control accounts for minimizing the variation of the considered paper properties along the machine direction. The starting point of our work is the simplified MPC tuning structure introduced in [12], in which the controller tuning problem is formulated as a two-degree-of-freedom (2-DOF) optimization problem. This framework has also been considered in [53], the target of which was to automatically determine the tuning parameters so that the performance requirements on worst-case overshoots and settling times are satisfied. Although fast responses with almost-optimal settling times satisfying the overshoot constraints can always be obtained by the three-step tuning algorithm introduced in [53], the responses could be oscillatory as the controller needs to behave aggressively to generate the smallest worst-case setting time without limiting the oscillations in the process output. These variations add to the wear and tear of the control valves and make the system more sensitive to actuator saturation [54], which not only downgrades the control performance but also increases the cost for maintenance. Based on the aforementioned considerations, the total variation is utilized for tuning method design in this work. Different from the overshoot,

[*] Parts of the results presented in this chapter are published in Industrial & Engineering Chemistry Research, vol. 54, no. 43, pp. 10811-10824, 2015.

the total variation is normally not intuitive to practitioners of the pulp and paper industry, and therefore the tuning tool should avoid users from directly specifying the requirement on total variation. In this regard, the specification on total variation is implicitly made according to the requirement on either the overshoot or other familiar time domain performance measures in this work. The main contributions are summarized as follows:

- A detailed robust stability analysis is given based on the small gain theorem and a tight robust stability condition is derived using a weighting function determined by the parametric uncertainty.
- Two flexible ways of specifying the output oscillation are proposed. In Section 2.4, the output oscillation is connected with the overshoot, which results in simple and fast tuning algorithms; in Section 2.5, the output oscillation is specified by the decay ratio, thus significantly reducing the conservativeness of tuning.
- By taking advantage of the unimodality and monotonicity properties of the conflicting time response measures, two efficient contour-line based parameter auto-tuning algorithms are proposed, based on which the robust time-domain specifications can be achieved.
- The efficiency and optimality of the proposed tuning algorithms are validated and analyzed for different uncertainty levels and robustness specifications on process models that are used for machine directional MPC of a paper machine at an industrial site.

2.2 Preliminaries and Problem Formulation

In this section, we first introduce the components in the following 2-DOF MPC control structure proposed in [12] and then describe the tuning problem to be solved.

2.2.1 Nominal model and model uncertainties

Based on the industrial practice, we consider the transfer function of the single-input, single-output real process G_p in Fig. 2.1 that has the FOPDT model

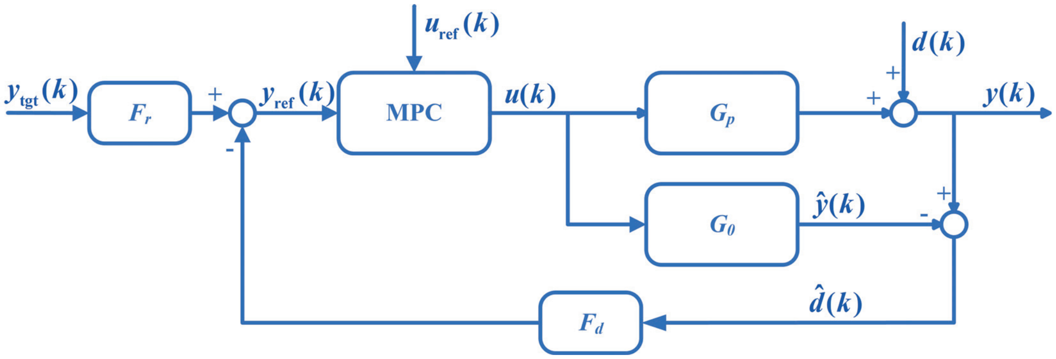


Figure 2.1: The 2-DOF MPC control system.

form:

$$G_p(s) = \frac{g}{T_p s + 1} e^{-t_d s}, \tag{2.1}$$

where g, T_p and t_d are the real process gain, time constant and time delay, respectively. As $G_p(s)$ cannot be exactly known in practice, a nominal model $G_0(s)$ is identified to approximate $G_p(s)$:

$$G_0(s) = \frac{g_0}{T_{p0} s + 1} e^{-t_{d0} s}. \tag{2.2}$$

The nominal model parameters g_0, T_{p0}, and t_{d0} are identified from the input and output data of the real process. However, it is inevitable that the identified model $G_0(s)$ is different from $G_p(s)$. To take into account the model mismatch, we introduce two types of model uncertainties in this work. One is the parametric uncertainty that refers to the mismatch in the model parameters, i.e.,

$$g \in [\underline{g}, \overline{g}],\ T_p \in [\underline{T}_p, \overline{T}_p], t_d \in [\underline{t}_d, \overline{t}_d]. \tag{2.3}$$

Using the parametric uncertainty, a set of possible perturbed plant models is denoted as

$$\Pi_1 := \left\{ G_p(s) : g \in [\underline{g}, \overline{g}], T_p \in [\underline{T}_p, \overline{T}_p], t_d \in [\underline{t}_d, \overline{t}_d] \right\}. \tag{2.4}$$

The other type of uncertainty we considered here is the multiplicative uncertainty, which is usually employed for robust stability analysis. For the multiplicative uncertainty, the set of possible perturbed plant models can be denoted as:

$$\Pi_2 := \{ G_0(s)(1 + W(s)\Delta(s)) : |\Delta(j\omega)| \le 1, \forall \omega \}, \tag{2.5}$$

where $\Delta(s)$ is a stable transfer function and $W(s)$ is a weighting function.

Note that the parametric uncertainty is easier for plant engineers to understand and thus it will be used throughout this study, while the multiplicative uncertainty will be only considered in the robust stability analysis. It is worthwhile mentioning that Π_1 and Π_2 are not necessary to be equal. However, by designing the weighting function $W(s)$, we may approximate Π_1 by Π_2 so that the robust stability condition can be derived when the parametric uncertainty is specified. Note that given an FOPDT model and a bounded parametric uncertainty, the weighting function $W(s)$ can always be calculated to connect Π_1 and Π_2, and the detailed discussion is shown in Section 2.3. In practice, this is highly desired by end-users with limited knowledge of robust control theory.

2.2.2 MPC and 2-DOF tuning

The MPC controller in Fig. 2.1 basically solves the following quadratic programming problem

$$\begin{aligned} \min_{\Delta U} \quad & J = \|\hat{Y} - Y_{\mathrm{ref}}\|_{Q_1} + \|\Delta U\|_{Q_2} + \|U - U_{\mathrm{ref}}\|_{Q_3} \\ \text{s.t.} \quad & \hat{x}(k+i) = A^i \hat{x}(k) + \sum_{j=1}^{\min\{\mathrm{H_u}, i\}} A^{i-j} B \Delta u(k+j-1), \\ & \hat{y}(k+i) = C\hat{x}(k+i), \ \text{for } i = 1, 2, \ldots, \mathrm{H_p}, \end{aligned}$$

where

$\hat{x}(k)$**:** estimated states at time k;

$\hat{y}(k+i)$**:** estimated controlled variables at time $k+i$;

$\Delta u(k)$**:** control move at time k;

$(A, B, C, 0)$**:** discrete-time state space realization of (2.2);

$\mathrm{H_u}$**:** control horizon;

$\mathrm{H_p}$**:** prediction horizon;

$\hat{Y}$= $\left[\hat{y}^T(k+1), \ \hat{y}^T(k+2), \cdots, \hat{y}^T(k+\mathrm{H_p})\right]^T$;

Y_{ref}**:** reference signal vector of $\hat{Y}$;

$\Delta U=\ \Big[\Delta u^T(k),\ \Delta u^T(k+1),\ \cdots,\ \Delta u^T(k+\mathrm{H_u}-1)\Big]^T;$

$$U=\ \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} u(k-1)+\begin{bmatrix}1 & 0 & \cdots & 0\\1 & 1 & \ddots & \vdots\\\vdots & \ddots & \ddots & 0\\1 & \cdots & 1 & 1\end{bmatrix}\Delta U;$$

U_{ref}**:** reference signal vector of U;

$Q_1,\ Q_2,\ Q_3$**:** diagonal penalty matrices;

$\|X\|_Q$**:** X^TQX.

In addition to the MPC controller, filters F_r and F_d are also an important part of the closed-loop control system in Fig. 2.1. These filters are respectively used for filtering the output target, $y_{tgt}(k)$, and the estimated disturbance, $\hat{d}(k) := y(k) - \hat{y}(k)$. With the filtered signals, we generate the reference trajectory based on

$$Y_{\mathrm{ref}}(k) = F_r y_{tgt}(k) - F_d\hat{d}(k).$$

Here F_r and F_d are the time domain implementations of $f_r(z)$ and $f_d(z)$, according to $y_{\mathrm{ref}}(z) = f_r(z)y_{tgt}(z) - f_d(z)\hat{d}(z)$, where $f_r(z)$ and $f_d(z)$ are the so-called reference tracking filter and disturbance rejection filter:

$$f_r(z) = \frac{b_r z^{-1}}{1 - a_r z^{-1}} z^{-T_{d0}},\ f_d(z) = \frac{b_d z^{-1}}{1 - a_d z^{-1}} z^{-T_{d0}}, \tag{2.6}$$

where $a_r = e^{-\frac{\Delta T}{\lambda T_{p0}}}$, $b_r = 1 - a_r$, $a_d = e^{-\frac{\Delta T}{\lambda_d T_{p0}}}$, $b_d = 1 - a_d$, ΔT is the sampling period and T_{d0} is the discretized version of t_{d0}. With these filters, the MPC performance can be adjusted by tuning λ and λ_d and setting $Q_1 = I$, $Q_2 = Q_3 = 0$, which simplifies the tuning problem [12]. Note that the new tuning parameters λ and λ_d can be used to achieve similar objectives as those by tuning Q_2 and Q_3. The reason is that λ and λ_d can be adjusted to filter the reference signal of the MPC so that there is no aggressive change in the reference trajectory, and correspondingly aggressive control signal can be avoided. Besides, the prediction horizon $\mathrm{H_p}$ and control horizon $\mathrm{H_u}$ are not considered as tuning parameters in this work and are set to fixed values according to the industrial experience [12].

2.2.3 Performance measures and tuning problem

In this work, we adopt the overshoot and settling time as the main performance measures for parameter tuning, since they are intuitive and well-suited for control performance evaluation for practitioners. As the performance concerned is for the whole set of perturbed systems in Π_1, the worst-case performance has to be utilized. The definitions of the worst overshoot and settling time are given as follows.

Definition 2.1 (Worst-case overshoot). *The worst-case overshoot* OS *of a set of step responses with the same final value is the maximum value in all responses minus the final value divided by the final value.*

Fig. 2.2 shows an illustration example of the worst-case overshoot.

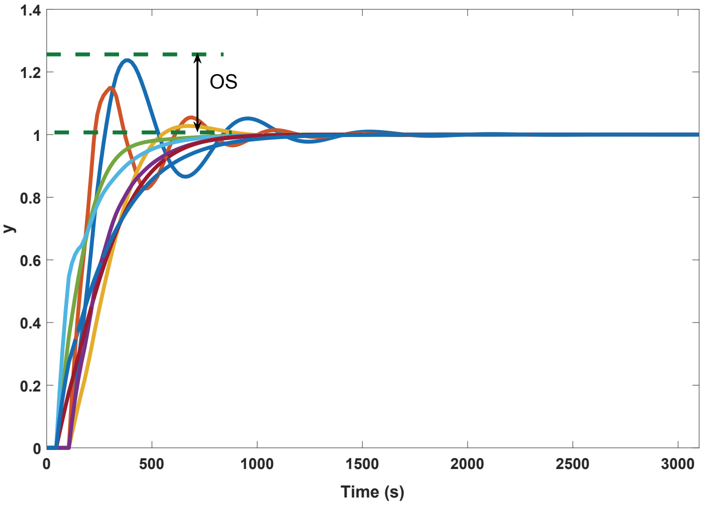


Figure 2.2: Illustration of worst-case overshoot OS.

Definition 2.2 (Worst-case settling time). *The worst-case settling time T_s of a set of step responses with the same final value is the maximum time, among all possible responses, to reach and stay within a range of a pre-specified percentage of the final value.*

Fig. 2.3 shows an illustration example of the worst-case settling time.

However, to efficiently characterize the OS and T_s is a non-trivial task. A conventional idea is to approximate the closed-loop system by a second-order

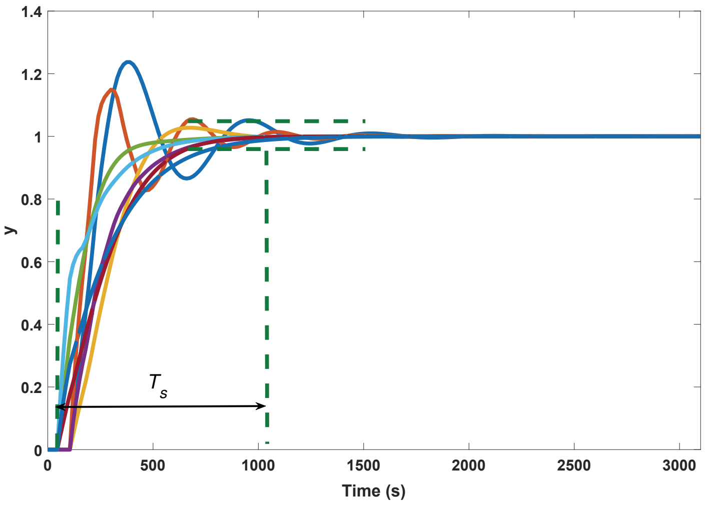


Figure 2.3: Illustration of worst-case settling time T_s.

system and compute the resultant OS and T_s using some existing formulas [44], but the approximation is a non-trivial task since the controlled system consists of an FOPDT model and an MPC and the approximation error is not predictable, even when the model parameters are exactly known. In this regard, as the step responses of a set of systems based on the parametric uncertainty (rather than a single system) are considered, the standard formulas for OS and T_s are not applicable here. In this work, we employ the heuristic approach introduced in [53], which computes the OS and T_s from the following 8 extreme-case systems:

$$\Pi_E := \left\{G_p(s) : g \in \{\underline{g}, \bar{g}\}, T_p \in \{\underline{T}_p, \bar{T}_p\}, t_d \in \{\underline{t}_d, \bar{t}_d\}\right\}, \tag{2.7}$$

where $x \in \{\underline{x}, \bar{x}\}$ means $x = \underline{x}$ or $x = \bar{x}$. As the extreme behavior of the step responses mostly happens at the extreme process parameters, the OS and T_s can be respectively approximated using the worst overshoot and the worst settling time of the extreme-case systems. The visualization technique utilized was developed based on industrial experience. Based on the analysis and extensive simulations in [53], the results obtained by (2.7) are acceptable since the worst-case OS and T_s can be calculated accurately and efficiently. As OS and T_s depend on the values of λ and λ_d in the tuning, we use the notations OS(λ, λ_d) and $T_s(\lambda, \lambda_d)$ to explicitly express these relationships when necessary.

In addition to the OS and T_s, another important time-domain performance measure is the total variation [55], which measures the output oscillation for general systems. It is defined mathematically as

$$tv := \sum_{k=0}^{\infty} |y(k+1) - y(k)|, \tag{2.8}$$

Fig. 2.4 shows an example of how to calculate the total variation. Here, we may ignore the effect of the sampling time and replace ∞ by n for ease of implementation, assuming that the system output converges to the final value within n steps. Note that in this work n is chosen according to $n = (\overline{T}_p + \bar{t}_d) \times 20$ with $\overline{T}_p$ and $\bar{t}_d$ defined in (2.4). Based on these simplifications, the worst-case total variation can be defined as below.

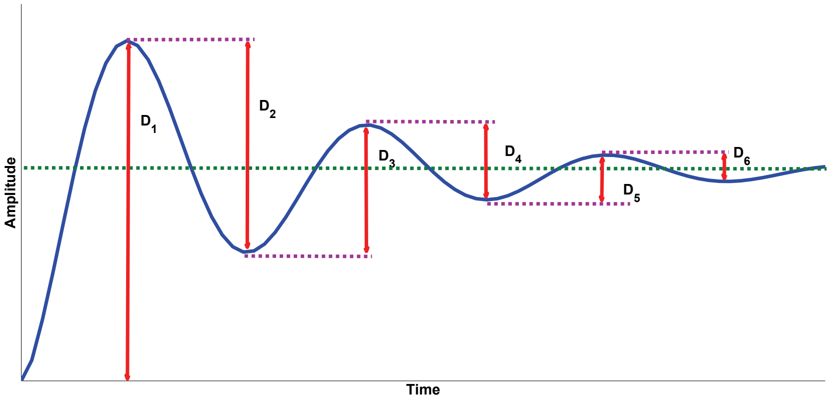


Figure 2.4: A graphical illustration example for total variation, where TV is equal to $\sum_i \mathrm{D}_i$.

Definition 2.3 (Worst-case total variation)**.**

$$\mathrm{TV} := \max_{G_p \in \Pi_1} \sum_{k=1}^{n} |y(k) - y(k-1)|. \tag{2.9}$$

Note that total variation is only calculated in the design stage and do not have to be calculated on-line with real data, as it is used as a performance index to avoid aggressive oscillations of the input and output signals in parameter tuning. To reflect the dependence on λ, λ_d of TV, we use the notation $\mathrm{TV}(\lambda, \lambda_d)$ to represent this relationship when necessary.

The tuning problem we are concerned is how to automatically determine λ and λ_d such that the robust stability is achieved and the system output can track its target with a small overshoot, settling time and output oscillation. However, multiple conflicts exist in achieving these objectives: a small overshoot often comes with a sluggish response, while a small settling time can be associated with an aggressive overshoot and total variation. Considering the tradeoff among different performance requirements, we tune λ and λ_d by minimizing the settling time while requiring OS and TV to lie in certain tolerable regions. Mathematically, it can be formulated in to the optimization problem as follows:

$$\begin{aligned} \min_{\lambda,\lambda_d} \quad & T_s(\lambda,\lambda_d) \\ \text{s.t.} \quad & \mathrm{OS}(\lambda,\lambda_d) \leq \mathrm{OS}^*, \\ & \mathrm{TV}(\lambda,\lambda_d) \leq \mathrm{TV}^*, \end{aligned} \tag{2.10}$$

where OS^* and TV^* refer to the specifications on $\mathrm{OS}(\lambda,\lambda_d)$ and $T_s(\lambda,\lambda_d)$, respectively. Fig. 2.5 shows a graphical illustration example of (2.10).

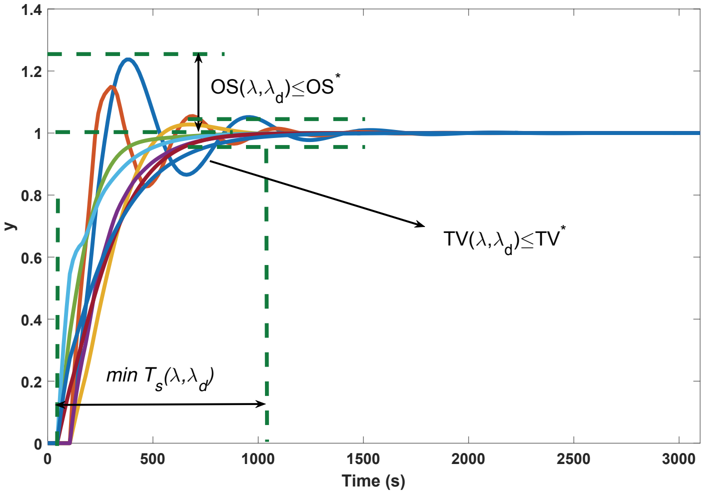


Figure 2.5: A graphical illustration example for the tuning problem.

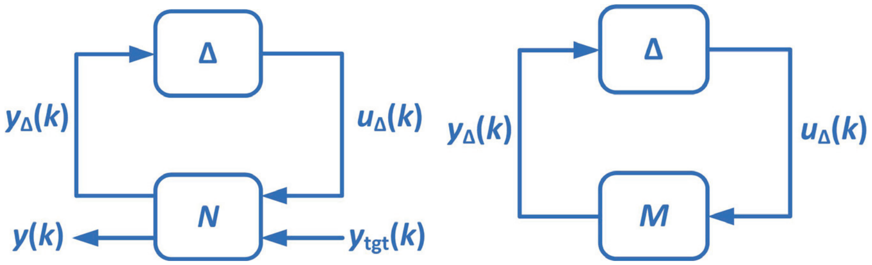
In addition to the difficulties in solving the problem at hand, the choice of TV^* is a non-intuitive task. Different from OS^* (which can be intuitively determined by the users based on their requirements, e.g., 10%), it is difficult for the end users to manually specify the appropriate value of TV^*. In this regard, TV^* is determined either automatically or based on the decay ratio (another well-understood control quality performance index in the pulp and paper industry), to maintain the user friendliness of the proposed tuning methods.

2.3 Robust Stability Conditions

Before tuning λ and λ_d with respect to the time-domain performance, we first investigate the robust stability of the closed-loop system, which is the top priority in the controller design. In Section 2.2, we have introduced two types of uncertainties: the parametric uncertainty and the multiplicative uncertainty. In this section, we will discuss how to build a bridge to connect them. Also, a tight robust stability condition based on parametric uncertainty will be given for the tuning of λ_d.

2.3.1 Robust stability condition for $G_p \in \Pi_2$

Considering the multiplicative uncertainty, the real process model is $G_p = G_0(s)(1 + W(s)\Delta(s)),\ |\Delta| \leq 1$. By pulling out Δ, the closed-loop system in Fig. 2.1 can be represented in a general form as shown in Fig. 2.6(a), see Zhou and Doyle [66].



(a) N-Δ (b) M-Δ

Figure 2.6: General representations for uncertain systems

The expression of N and the transfer function of the closed-loop system are as follows:

$$N = \left[\begin{array}{c|c} N_{11} & N_{12} \\ \hline N_{21} & N_{22} \end{array}\right], \tag{2.11}$$

$$F(N,\Delta) = N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12}. \tag{2.12}$$

As the stability of the closed-loop system only depends on N_{11}, we thus examine the M-Δ system (see Fig. 2.6(b), where $M = N_{11}$) instead of the N-Δ system. With some algebraic manipulations, we have

$$M = -(1 + F_{\text{MPC}}\underline{F_d^1} + F_{\text{MPC}}\underline{F_d^2}G_0)^{-1}F_{\text{MPC}}\underline{F_d^2}G_0W, \tag{2.13}$$

where F_{MPC} is the transfer function from $y_{ref}(k)$ to $u(k)$ and can be derived from the unconstrained MPC control law; $\underline{F_d^1}$ and $\underline{F_d^2}$ are the transfer functions from $u(k)$ to $y_{ref}(k)$ and $y(k)$ to $y_{ref}(k)$, respectively. For the M-Δ system, the system is robustly stable, if and only if,

$$\det(1 - M(j\omega)\Delta) \neq 0,\ \forall\omega, \forall|\Delta| \leq 1. \tag{2.14}$$

For SISO systems, it is equivalent to

$$\begin{aligned} &|M(j\omega)| < 1,\ \forall\omega, \forall|\Delta| \leq 1, \\ \Leftrightarrow &|W(j\omega)T(j\omega)| < 1,\ \forall\omega, \forall|\Delta| \leq 1 \end{aligned} \tag{2.15}$$

where $T(s) = \frac{1}{\lambda_d T_{p0}s+1}e^{-t_{d0}s}$ is the sensitivity function of the closed-loop system, which is obtained following some tedious but straightforward mathematical manipulations, and the readers can refer to [12, 51] for the detailed discussion.

2.3.2 Construction of $W(s)$ with $G_p \in \Pi_1$

The robust conditions in (2.15) are derived for $G_p \in \Pi_2$, which may not be applicable to the systems in Π_1. Fortunately, the freedom of choosing $W(s)$ provides us possibility to closely connect Π_1 and Π_2.

From $G_p \in \Pi_2$, Δ can be written as $W^{-1}(G_p - G_0)G_0^{-1}$, whose modulus is known to be less than 1. It implies that

$$|W| \geq |(G_p - G_0)G_0^{-1}|. \tag{2.16}$$

This suggests us a way to construct the weighting function W using the upper bound of $|(G_p - G_0)G_0^{-1}|$, which is known as the multiplicative error [38]. Note that the closer $|W|$ is to the upper bound of $|(G_p - G_0)G_0^{-1}|$ at all frequencies, the less conservative the robust condition is.

We now use the parametric uncertainty to find the upper bound of $|(G_p - G_0)G_0^{-1}|$. Without loss of generality, we can assume that,

$$\begin{aligned} g_0 - \Delta g \leq g \leq g_0 + \Delta g, \\ T_{p0} - \Delta T_p \leq T_p \leq T_{p0} + \Delta T_p, \\ t_{d0} - \Delta t_d \leq t_d \leq t_{d0} + \Delta t_d. \end{aligned}$$

Then, the multiplicative error becomes:

$$(G_p - G_0)G_0^{-1} = \frac{ge^{(t_{d0}-t_d)s}(T_{p0}s+1) - g_0(T_ps+1)}{g_0(T_ps+1)}. \tag{2.17}$$

By using Padé approximation, we can change (2.17) into a real-rational form as below,

$$\begin{aligned} &(G_p - G_0)G_0^{-1} \approx \\ &\frac{g(1+\frac{t_{d0}-t_d}{2}s)(T_{p0}s+1) - g_0(T_ps+1)(1-\frac{t_{d0}-t_d}{2}s)}{g_0(T_ps+1)(1-\frac{t_{d0}-t_d}{2}s)}. \end{aligned} \tag{2.18}$$

The upper bound of $|(G_p - G_0)G_0^{-1}|$ can be determined with the inspection of the denominator and the numerator of (2.18). For the numerator, we have

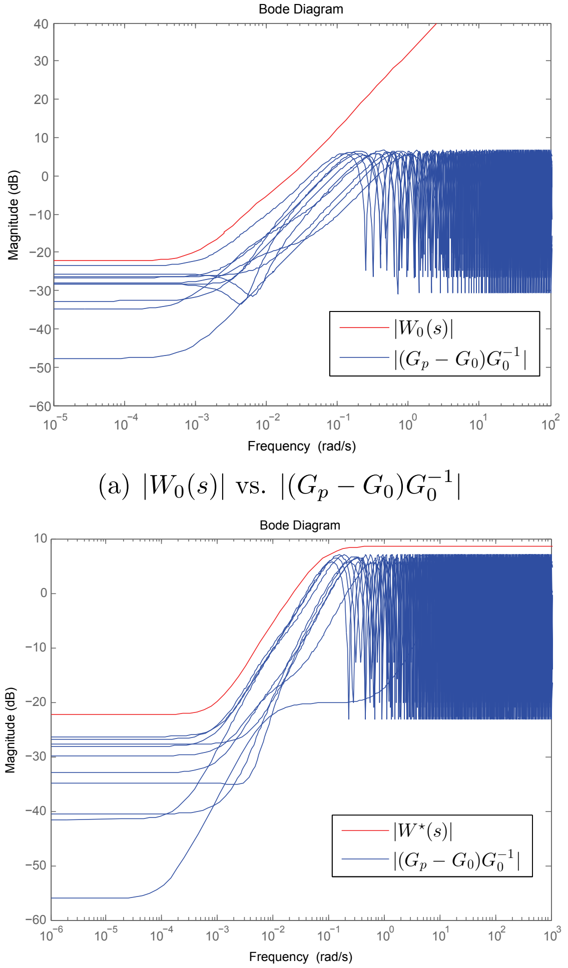
$$\begin{aligned} &|g(1+\frac{t_{d0}-t_d}{2}s)(T_{p0}s+1) - g_0(T_ps+1)(1-\frac{t_{d0}-t_d}{2}s)|^2 \\ &= [(g-g_0) + (gT_{p0} + g_0T_p)\frac{t_d - t_{d0}}{2}\omega^2]^2 + \\ &\quad [gT_{p0} - g_0T_p + (g+g_0)\frac{t_{d0}-t_d}{2})]^2\omega^2 \\ &\leq \{\Delta g + [(g_0+\Delta g)T_{p0} + g_0(T_{p0} + \Delta T_p)]\omega^2\}^2 + \\ &\quad [\Delta g(T_{p0} + 0.5\Delta t_d) + g_0(\Delta T_p + \Delta t_d)]^2\omega^2. \end{aligned} \tag{2.19}$$

For the denominator of (2.18), we get

$$\begin{aligned} &|g_0(T_ps+1)(1-\frac{t_{d0}-t_d}{2}s)|^2 \\ &= g_0^2[1+(T_p\omega)^2][1+(\frac{t_{d0}-t_d}{2}\omega)^2] \\ &\geq g_0^2[1+(T_{p0}-\Delta T_p)^2\omega^2]. \end{aligned} \tag{2.20}$$

Thus, the expression of $W(s)$ can be constructed as below:

$$\begin{aligned} W_0(s) = &\frac{\Delta g + [\Delta g(T_{p0} + 0.5\Delta t_d) + g_0(\Delta T_p + \Delta t_d)]s}{g_0[(T_{p0} - \Delta T_p)s + 1]} \\ &- \frac{[(g_0+\Delta g)T_{p0} + g_0(T_{p0}+\Delta T_p)]0.5\Delta t_d s^2}{g_0[(T_{p0}-\Delta T_p)s+1]}, \end{aligned} \tag{2.21}$$



(a) $|W_0(s)|$ vs. $|(G_p - G_0)G_0^{-1}|$

(b) $|W(s)^{\star}|$ vs. $|(G_p - G_0)G_0^{-1}|$

Figure 2.7: The bode magnitude plot of $W_0(s)$, $W^{\star}(s)$ and multiplicative errors

which satisfies $|W_0| \geq |(G_p - G_0)G_0^{-1}|$. However, we may observe that $W_0(s)$ shown above is a bit conservative, because it has large values in high frequencies, see Fig. 2.7(a). To fix this issue, we multiply $(1 - as)$ in the denominator to get $W^{\star}(s) = \frac{1}{1-as}W_0(s)$, which results in

$$|W^{\star}(\infty)| = |\frac{[(g_0 + \Delta g)T_{p0} + g_0(T_{p0} + \Delta T_p)]0.5\Delta t_d}{ag_0(T_{p0} - \Delta T_p)}|. \tag{2.22}$$

On the other hand, we see that

$$\begin{aligned} &|(G_p(\infty) - G_0(\infty))G_0^{-1}(\infty)| \\ &\approx |\frac{gT_{p0} + g_0T_p}{g_0T_p}| \leq |\frac{[(g_0 + \Delta g)T_{p0} + g_0(T_{p0} + \Delta T_p)]}{g_0(T_{p0} - \Delta T_p)}| \end{aligned} \tag{2.23}$$

To have $|W^\star(\infty)| \geq |(G_p(\infty) - G_0(\infty))G_0^{-1}(\infty)|$, we may let $a = 0.5\Delta t_d$ and therefore

$$W^\star(s) = \frac{1}{1 - 0.5\Delta t_d s} W_0(s). \tag{2.24}$$

Fig. 2.7(b) shows a bode magnitude plot of an example system, where blue curves are corresponding to the multiplicative errors of perturbed systems in Π_1. It is seen that the proposed $|W(s)|$ is indeed a tight bound of $|(G_p - G_0)G_0^{-1}|$ over all frequencies. Note that given the parametric uncertainty and FOPDT model, a quantitative evaluation of the conservativeness is normally difficult to perform explicitly. Alternatively, the conservativeness of the proposed results is tested based on extensive simulations and the obtained bound is normally observed to be tight (see Fig. 2.7 for some typical results).

Based on (2.15) and (2.24), a tight robust stability condition on λ_d can be derived:

$$\lambda_d \geq \frac{\sqrt{1 - |W^\star(j\omega)|^2}}{T_{p0}\omega|W^\star(j\omega)|} := \underline{\lambda}_d^\star, \quad \forall\omega. \tag{2.25}$$

$\underline{\lambda}_d^*$ will serve as the minimum allowed value of λ_d in the following tuning analysis.

2.4 Efficient Tuning with Time-domain Performance

In this section, we present the first approach to the optimization problem in (2.10). The value for overshoot requirement OS^* is easy to specify, e.g., 10%; however, the specification of TV^* is not straightforward. Since the constraint on total variation is utilized to limit the potential oscillations in the system output, it is possible to link the value of TV^* with OS^*, for which we have the following relationship.

Proposition 2.1. *If* $\mathrm{TV}(\lambda, \lambda_d) \leq \mathrm{TV}^*$ *and* $\mathrm{OS}(\lambda, \lambda_d) > 0$*, then*

$$\mathrm{OS}(\lambda, \lambda_d) \leq (\mathrm{TV}^* - 1)/2.$$

Proof: For the unit step response $y(k)$ of the system in Fig. 2.1, let

$$y_{\max} := \max_{k\in\{0,\dots,n\}} y(k),$$
$$n_{\max} := \arg \max_{k\in\{0,\dots,n\}} y(k).$$

According to the definition of the worst-case total variation, we have

$$\begin{aligned}
&\mathrm{TV}(\lambda, \lambda_d) \\
&= \max_{G_p\in\Pi_E} \sum_{k=1}^{n} |y(k) - y(k-1)| && (2.26)\\
&\geq \max_{G_p\in\Pi_E} \left(\left| \sum_{k=1}^{n_{\max}} [y(k) - y(k-1)] \right| \right. \\
&\qquad \left. + \left| \sum_{k=n_{\max}+1}^{n} [y(k) - y(k-1)] \right| \right) && (2.27)\\
&= \max_{G_p\in\Pi_E} |y_{\max} - y(0)| + |y(n) - y_{\max}| && (2.28)\\
&= \max_{G_p\in\Pi_E} (|y_{\max} - y(n) + y(n) - y(0)| \\
&\qquad +|y_{\max} - y(n)|) && (2.29)\\
&= \max_{G_p\in\Pi_E} 2|y_{\max} - y(n)|/|y(n)| + |y(n) - y(0)| && (2.30)\\
&= 1 + 2 \max_{G_p\in\Pi_E} |y_{\max} - y(n)|/|y(n)| && (2.31)\\
&= 1 + 2\mathrm{OS}(\lambda, \lambda_d), && (2.32)
\end{aligned}$$

where inequality (2.27) follows from the triangle inequality, equation (2.28) follows from the definition of $y_{\max}$ and $n_{\max}$, equation (2.30) follows from the fact that $y_{\max} - y(n) \geq 0$, $y(n) - y(0) > 0$ and $y(n) = 1$, equation (2.31) is due to $y(0) = 0$, and equation (2.32) follows from the definition of the worst-case overshoot. The conclusion follows from $\mathrm{TV}(\lambda, \lambda_d) \leq \mathrm{TV}^*$.

The above result shows that given the specification on OS^*, the specification on TV^* can be chosen as $1+2\mathrm{OS}^*$ to ensure a smooth response. Also, this helps simplify the problem in (2.10), as by determining $\mathrm{TV}^* = 1 + 2\mathrm{OS}^*$, the specification on overshoot can be automatically satisfied. This also reduces the requirement of user's knowledge on the process, as no other information is needed to specify TV^*, except for the requirement on overshoot OS^*, which is normally familiar to the end-users of commercial quality control software.

Based on the above discussion, the problem in (2.10) now reduces to

$$\begin{aligned} \min_{\lambda,\lambda_d} \;& T_s(\lambda,\lambda_d) \\ s.t. \;& \mathrm{TV}(\lambda,\lambda_d) \leq \mathrm{TV}^*, \end{aligned} \tag{2.33}$$

where $\mathrm{TV}^* = 1 + 2\mathrm{OS}^*$.

2.4.1 Empirical monotonicity properties of TV with respect to λ and λ_d

Similar to the approach in [53], the empirical unimodality and monotonicity properties of TV with respect to λ and λ_d are investigated and employed to solve the robust tuning problem, due to the lack of analytical expression of TV, even for standard second-order systems.

Since λ controls the speed of the output response, a larger value of λ leads to smoother response and thus a smaller value of total variation. In this way, $\mathrm{TV}(\lambda,\lambda_d)$ can be empirically considered as a monotonically decreasing function of λ. Note that this property is determined by the 2DOF structure of the control system and is generally held in the considered paper-making processes. This property is further verified through numerical simulations, and a typical plot of the monotonicity relationship is shown in Fig. 2.8. Note that the monotonicity property is very useful in solving the tuning problem. As analytical expressions of the performance indices are not known, given this property, the search of the tuning parameters can be achieved based on simple and fast optimization methods, like the bisection method in this work. Otherwise, we need to use intelligent optimization methods which normally require more computational time.

2.4.2 The contour-line optimal tuning algorithm

In this section, we introduce the contour-line based tuning algorithm, which is proposed according to the monotonicity and unimodality properties (Unimodality here means that there is only one maximum/minimum in the defined interval of a function). Here we assume the user-specified total variation (or the equivalent specification on overshoot based on Proposition 2.1) is relatively small such that, for the optimal solution, the constraint in problem (2.33) is active; this is intuitive in that for small total variation specifications, aggressive λ-parameters would be chosen to achieve small settling time, but

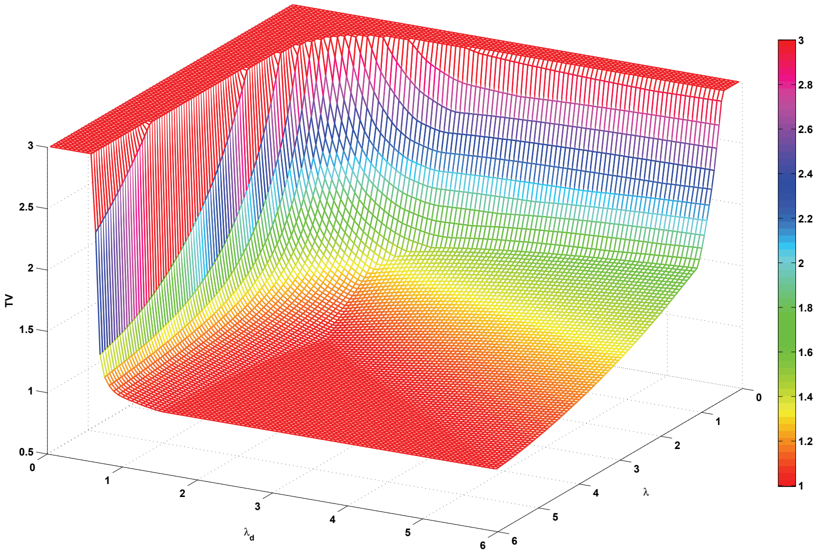


Figure 2.8: Empirical monotonicity properties of worst-case total variation with respect to λ and λ_d.

these may result in increased overshoot or total variation. The aforementioned assumption is mainly introduced to reduce the computational burden required to solve the tuning problem, as the line-search procedure to find the value of λ that corresponds to the smallest T_s given a fixed value of λ_d and $\mathrm{TV}(\lambda, \lambda_d) \leq \mathrm{TV}^*$ can be avoided.

Algorithm 1 Find $\lambda^c(\mathrm{TV}^*, \lambda_d)$ and $T_s^c(\mathrm{TV}^*, \lambda_d)$.

```
Input $\mathrm{TV}^*$, $\lambda_d$ and the uncertainty intervals $[\underline{g}, \overline{g}]$, $[\underline{T}_p, \overline{T}_p]$ and $[\underline{t}_d, \bar{t}_d]$;
Input $\epsilon$;                                         $\triangleright$ $\epsilon = 0.1$ by default
$\underline{\lambda} \leftarrow 0.02$; $\bar{\lambda} \leftarrow 100$;
while $\bar{\lambda} - \underline{\lambda} > \epsilon$ do
    $\lambda \leftarrow (\bar{\lambda} + \underline{\lambda}) \times 0.5$;
    Numerically evaluate the total variation $\mathrm{TV}(\lambda, \lambda_d)$ based on (2.9);
    if $\mathrm{TV}(\lambda, \lambda_d) - \mathrm{TV}^* > 0$ then
        $\underline{\lambda} \leftarrow \lambda$;
    else
        $\bar{\lambda} \leftarrow \lambda$;
    end if
end while
$\lambda^c(\mathrm{TV}^*, \lambda_d) \leftarrow (\bar{\lambda} + \underline{\lambda})/2$;
$T_s^c(\mathrm{TV}^*, \lambda_d) \leftarrow T_s(\lambda^c, \lambda_d)$;
end
```

Based on the above analysis, the optimization problem can be converted to

$$\begin{aligned} &\min_{\lambda,\lambda_d} \; T_s(\lambda,\lambda_d) \\ &s.t. \;\; \mathrm{TV}(\lambda,\lambda_d) = \mathrm{TV}^*. \end{aligned} \tag{2.34}$$

In this regard, it suffices to minimize the settling time $T_s(\lambda, \lambda_d)$ along the contour line $\mathrm{TV}(\lambda, \lambda_d) = \mathrm{TV}^*$. In order to solve this problem, we propose two algorithms: the first algorithm (Algorithm 1) finds the corresponding λ (which is denoted as λ^c in the following context) on the contour given a fixed λ_d, while the second algorithm (Algorithm 2) is introduced to search for the optimal λ_d that provides the smallest settling time, with the help of Algorithm 1. The algorithms are further interpreted in details in the following. Note that as we do not have analytical expressions for the performance indices utilized in the considered optimization problem, it may not be easily handled by a standard optimization algorithm. Besides, should an intelligent optimization algorithm (e.g., genetic algorithm) be utilized, the computational time would increase a lot, which would downgrade the user-friendliness of the proposed tuning algorithm.

Interpretations for Algorithm 1: As $\mathrm{TV}(\lambda, \lambda_d)$ is a monotonic function of λ for a fixed λ_d, the search of λ^c that leads to any given value of TV^* for a fixed λ_d can be achieved by a bisection search. To avoid notational confusion, we use $T_s^c(\mathrm{TV}^*, \lambda_d)$ to represent the dependence of the worst-case settling time on TV^* and λ_d. Besides, the upper and lower bounds of λ are chosen based on the industrial experience and it is worth noting that the computation time of the algorithm will increase if a larger search region (determined by the upper and lower bounds) is considered.

Algorithm 2 Tuning of λ and λ_d

```
Input the uncertainty intervals $[\underline{g}, \overline{g}]$, $[\underline{T}_p, \overline{T}_p]$ and $[\underline{t}_d, \overline{t}_d]$;
Input the overshoot specification $\mathrm{OS}^*$;
Calculate the total variation specification according to $\mathrm{TV}^* = 1 + 2\mathrm{OS}^*$;
Input $\epsilon$;                                              ▷ $\epsilon = 0.1$ by default
Calculate $\lambda_d^\star$ based on (2.25);
$\underline{\lambda}_d \leftarrow \underline{\lambda}_d^\star$; $\bar{\lambda}_d \leftarrow 100$;
while $\bar{\lambda}_d - \underline{\lambda}_d > \epsilon$ do
    $\lambda_{d1} \leftarrow \underline{\lambda}_d + (\bar{\lambda}_d - \underline{\lambda}_d) \times 0.382$;
    $\lambda_{d2} \leftarrow \underline{\lambda}_d + (\bar{\lambda}_d - \underline{\lambda}_d) \times 0.618$;
    Numerically evaluate the settling times $T_s^c(\mathrm{TV}^*, \lambda_{d1})$ and $T_s^c(\mathrm{TV}^*, \lambda_{d2})$
  based on Algorithm 1;
    if $T_s^c(\mathrm{TV}^*, \lambda_{d1}) > T_s^c(\mathrm{TV}^*, \lambda_{d2})$ then
        $\underline{\lambda}_d \leftarrow \lambda_{d1}$;
    else
        $\bar{\lambda}_d \leftarrow \lambda_{d2}$;
    end if
end while
$\lambda_d \leftarrow (\bar{\lambda}_d + \underline{\lambda}_d)/2$;
$\lambda \leftarrow \lambda^c(\mathrm{TV}^*, \lambda_d)$;
end
```

Interpretations for Algorithm 2: The assumption utilized here is that $T_s^c(\mathrm{TV}^*, \lambda_d)$ is a unimodal function of λ_d; the underlying cause is that λ_d controls the stability of the system: a small λ_d leads to an aggressive and oscillatory response, while a large λ_d leads to a sluggish response. In this way, the algorithm uses golden search to find the optimal λ_d that achieves the smallest worst-case settling time; the corresponding value of λ can be obtained through Algorithm 1. Note that the golden search method is used due to the unimodal property of $T_s^c(\mathrm{TV}^*, \lambda_d)$.

Given the optimization problem in (2.34), Algorithm 2 searches for the λ_d that minimizes the settling time T_s. Also, Algorithm 2 uses Algorithm 1 to calculate settling time T_s for each λ_d value. In calculating T_s, Algorithm 1 determines the value of λ so that $\mathrm{TV} = \mathrm{TV}^*$ for a fixed λ_d. In this way, Algorithm 2 finds λ and λ_d such that a small settling time can be obtained while the constraint $\mathrm{TV} = \mathrm{TV}^*$ is satisfied, with the help of Algorithm 1.

2.5 Determining TV* from the Decay Ratio

In the previous section, we have presented two efficient auto-tuning algorithms and we have connected TV* with OS* to automatically determine TV*. Although it gives a fast and simple tuning procedure and OS $\leq$ OS* is guaranteed, it may involve conservativeness in the tuned OS due to the fact that the oscillation is overlooked when the output response is approaching the steady state. This section aims at providing an alternative heuristic user-friendly approach to specify TV* through decay ratio to reduce the conservativeness.

2.5.1 Determine TV* using decay ratios

For stable second-order linear systems, the output oscillation is often measured by the decay ratio (denoted as DR), which is defined as the ratio between two consecutive maxima of the output after a step change in the set point and lies in $(0, 1)$, see Fig. 2.9. From the engineering perspective, the response of the system in Fig. 2.1 can be approximated by that of a second-order system. Therefore it is reasonable to set TV* equal to the total variation of a second-order linear system with a maximum allowed decay ratio DR*, which is specified by users with process knowledge. Fig. 2.10 sketches the procedure to specify the TV*.

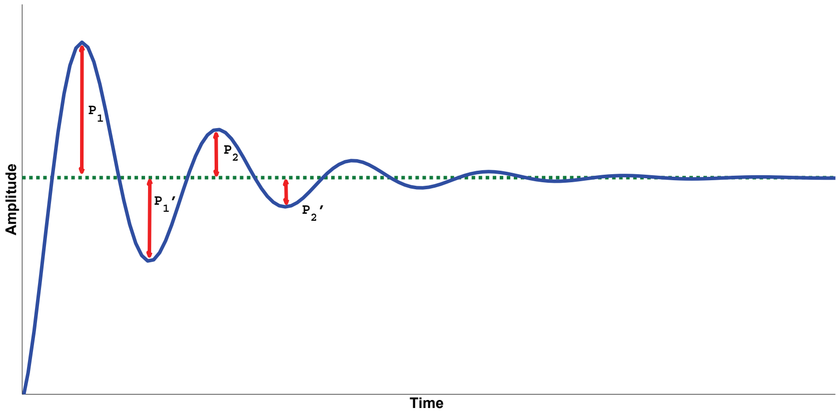


Figure 2.9: Illustration of the decay ratio for the response of a second-order system, where DR is equal to P_2/P_1.

In this section, it is assumed that DR* is specified by users. When users have limited knowledge of the process or have no particular requirement on the oscillation specification, DR* will be set to 1/4 by default, since 'one

User's knowledge
DR*
TV*

Figure 2.10: The specification of TV*

quarter decay ratio' is normally utilized as the design criterion for controller tuning [42]. The following formula is utilized to calculate TV* from DR*:

$$\mathrm{TV}^* = 1 + \frac{3\mathrm{OS}^*}{1 - \mathrm{DR}^*}. \tag{2.35}$$

The above formula is constructed using the total variation of a second-order system with DR=DR*. The total variation of the system in Fig. 2.10 can be approximately calculated as

$$\begin{aligned} tv &= 1 + 2(\mathrm{P}_1 + \mathrm{P}_2 + \cdots) + 2(\mathrm{P}_1' + \mathrm{P}_2' + \cdots) \\ &= 1 + 3(\mathrm{P}_1 + \mathrm{P}_2 + \cdots) \\ &= 1 + 3\mathrm{P}_1(1 + \mathrm{DR} + \mathrm{DR}^2 + \cdots) \\ &= 1 + \frac{3\mathrm{P}_1}{1 - \mathrm{DR}}, \end{aligned} \tag{2.36}$$

where we empirically assume that each lower peak amplitude is a half of the previous upper peak amplitude, considering the 'one quarter decay ratio' criterion (namely, ${\mathrm{P}_1}'/\mathrm{P}_1 = 1/2$ and $\mathrm{P}_2/\mathrm{P}_1' = 1/2$, which gives $\mathrm{P}_2/\mathrm{P}_1 = 1/4$). Since the amplitude of the first upper peak P_1 is also equal to the overshoot of the system, we may replace it by OS* if we let DR=DR*. Then, we have the formula as shown in (2.35). Note that P_1 is replaced by the maximum overshoot OS* in (2.35) because of two reasons: 1) in the case of DR=DR*, the overshoot is intuitively close or equal to OS*; 2) the choice of OS* maximizes the tolerance on the oscillation, which reduces the conservativeness of tuning.

Algorithm 3 Find λ^c and T_s^c with TV^* specified by (2.35)

```
Input $\mathrm{TV}^*$, $\lambda_d$ and the uncertainty intervals $[\underline{g}, \overline{g}]$, $[\underline{T}_p, \overline{T}_p]$ and $[\underline{t}_d, \overline{t}_d]$;
Input $\epsilon$;                                                  ▷ $\epsilon = 0.1$ by default
$\underline{\lambda} \leftarrow 0.02$; $\bar{\lambda} \leftarrow 100$;
whichInUse =‘TV’;
**while** $\bar{\lambda} - \underline{\lambda} > \epsilon$ **do**
    $\lambda \leftarrow (\bar{\lambda} + \underline{\lambda}) \times 0.5$;
    Numerically evaluate $\mathrm{TV}(\lambda, \lambda_d)$ and $\mathrm{OS}(\lambda, \lambda_d)$ based on (2.9) and Definition 2.1, respectively;
    **switch** whichInUse
        **case** ‘TV’
            **if** $\mathrm{TV}(\lambda, \lambda_d) - \mathrm{TV}^* > 0$ **then**
                $\underline{\lambda} \leftarrow \lambda$;
            **else**
                **if** $\mathrm{OS} < \mathrm{OS}^*$ **then**
                    $\bar{\lambda} \leftarrow \lambda$;
                **else**
                    whichInUse =‘OS’;
                    $\underline{\lambda} \leftarrow \lambda$;
                **end if**
            **end if**
        **case** ‘OS’
            **if** $\mathrm{OS}(\lambda, \lambda_d) - \mathrm{OS}^* > 0$ **then**
                $\underline{\lambda} \leftarrow \lambda$;
            **else**
                $\bar{\lambda} \leftarrow \lambda$;
            **end if**
    **end switch**
**end while**
$\lambda^c(\mathrm{TV}^*, \lambda_d) \leftarrow (\bar{\lambda} + \underline{\lambda})/2$;
$T_s^c(\mathrm{TV}^*, \lambda_d) \leftarrow T_s(\lambda^c, \lambda_d)$;
**end**
```

2.5.2 A switching tuning mechanism

The above approach to specify TV^* provides the users more insight and control of the output oscillation, but it also opens up the possibility that the tuned OS is greater than OS^*. To avoid this phenomenon, we introduce a switching tuning mechanism in the tuning of λ^c, which guarantees $\mathrm{OS} \leq \mathrm{OS}^*$. The tuning procedure is given in Algorithm 3.

It works in the following way: when $\mathrm{TV}(\lambda, \lambda_d)$ reaches TV^*, if the associated $\mathrm{OS}(\lambda, \lambda_d)$ at this time is smaller than OS^*, then the algorithm stops; otherwise, it continues to tune λ until $\mathrm{OS}(\lambda, \lambda_d)$ approaches OS^*; note that as λ has a similar monotonic relationship with both TV and OS (i.e., the increase of λ will decrease TV and OS), the tuning the OS here will not violate the constraint on TV. Clearly, if $\mathrm{OS}(\lambda, \lambda_d)$ is always less than OS^* in the tuning process, the tuning of λ^c is along the same line of that in Algorithm 1. Note that based on the structure of the 2-DOF MPC, if λ and λ_d are sufficiently large, then $\mathrm{TV} \to 1$ and $\mathrm{OS} \to 0$, so the feasible solution set for the tuning problem is non-empty if the upper bounds for λ and λ_d are sufficiently large. In this work, we have set the upper bounds of λ and λ_d to be 100, which normally can guarantee a feasible solution.

Based on Algorithm 3, we can update Algorithm 2 by modifying its line 10 to "Numerically evaluate the settling times $T_s^c(\mathrm{TV}^*, \lambda_{d1})$ and $T_s^c(\mathrm{TV}^*, \lambda_{d2})$ based on Algorithm 3;". Hereafter, this new version of Algorithm 2 will be named as Algorithm 4 for simplicity.

2.5.3 Discussion

In this work, two Algorithms (namely, Algorithm 2 and Algorithm 4) are proposed to solve the problem in (2.10), providing the end users alternative options to determine the values of the tuning parameters for satisfactory performance. Algorithm 2 determines the worst-case total variation specification automatically by exploiting the relationship between the overshoot and total variation of a step response. Given the same specification on worst-case overshoot, Algorithm 4 potentially leads to a smaller settling time, as the corresponding specification on worst-case total variation made based on (2.35) is larger than that of Algorithm 2; however, the sacrifice is that end users need to manually specify the specification on decay ratio and the relationship between the overshoot specification and the total variation specification is heuristic rather than rigorously established. Given the same specification on worst-case total variations, Algorithm 2 inherits a larger opportunity of finding the optimal solution as the worst-case settling time is directly minimized, while Algorithm 4 is equipped with a switching tuning mechanism, which has a potential effect of deviating the tuning results from the optimal solutions.

2.6 Application Examples

To illustrate the performance of the proposed tuning algorithms, extensive evaluations and comparisons are presented in this section by applying the proposed results to examples extracted from real applications of machine directional paper machine control. The details of the papermaking process can be found in [13].

First, we evaluate the proposed stability condition, test the optimality of the proposed tuning algorithms and compare the tuning performance based on Algorithms 2 and 4. The following nominal system is considered,

$$G_0(s) = \frac{0.0135}{60s+1} e^{-90s}, \tag{2.37}$$

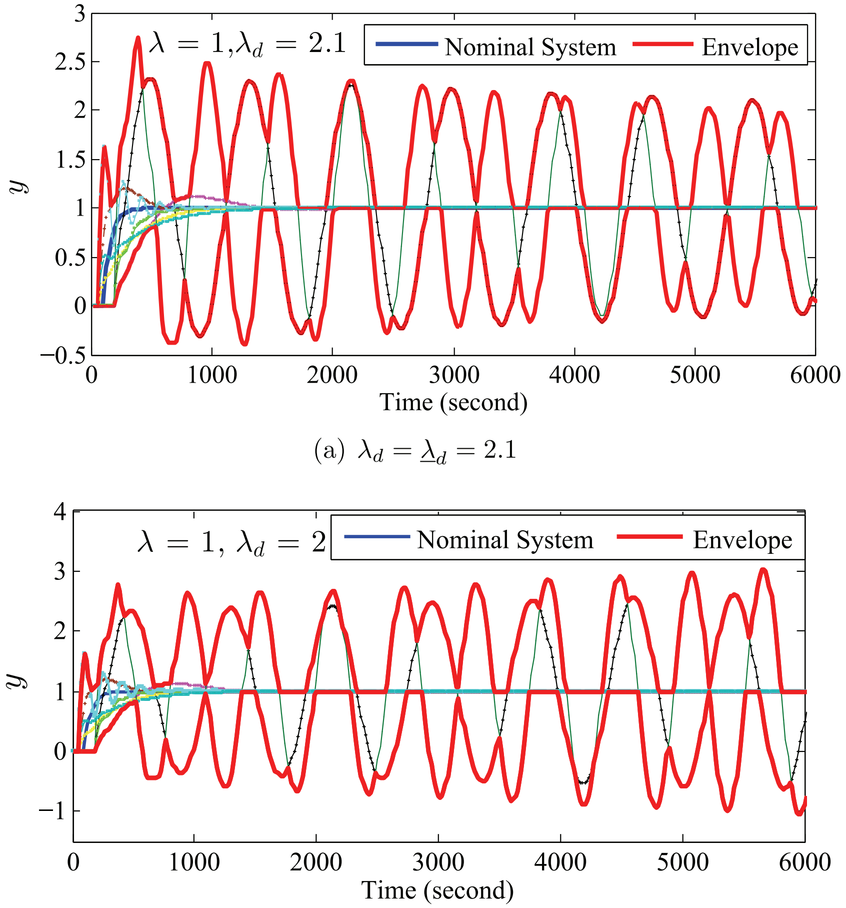
which models a papermaking process from stock to conditioned weight. The model was identified using an advanced industrial control software package and used by an MPC controller for a real paper machine. The prediction and control horizons are set to $\mathrm{H_p} = 42$ and $\mathrm{H_u} = 20$, respectively. The uncertainty level is defined as $[-\underline{r}\%, \bar{r}\%]$, which means that the real model parameters are chosen from the following intervals:

$$\begin{aligned} T_p &\in [(1-\underline{r}\%)T_{p0}, (1+\bar{r}\%)T_{p0}], \\ t_d &\in [(1-\underline{r}\%)t_{d0}, (1+\bar{r}\%)t_{d0}], \\ g &\in [(1-\underline{r}\%)g_0, (1+\bar{r}\%)g_0]. \end{aligned} \tag{2.38}$$

A large uncertainty level [-40%, 100%] is used throughout the example. Based on the robust stability condition in (2.25), $\underline{\lambda}_d$ is calculated and it equals to 2.1. Fig. 2.11 shows the step response of the eight extreme-case systems under different λ_d's. The envelopes of step responses of the uncertain systems are shown in the red solid lines. In Fig. 2.11(a), λ_d is set to $\underline{\lambda}_d$ and from the envelopes we can see that the closed loop system is still robustly stable. By contrast, when λ_d is slightly decreased (from 2.1 to 2.0) in Fig. 2.11(b), robust stability no longer holds. This implies that the obtained $\underline{\lambda}_d$ is indeed a tight lower bound that guarantees robust stability.

It is worth noting that the envelope responses from $\lambda_d = \underline{\lambda}_d$ can be aggressive. Thus, in proposed algorithms, we further tune λ_d for better performance.

Next, we proceed to test the optimality of Algorithm 2 for different worst-case total variation specifications, where a brutal search is applied to identify the near-optimal solution to the problem in (2.33). Note that the optimality



(a) $\lambda_d = \underline{\lambda}_d = 2.1$

(b) $\lambda_d = 2$

Figure 2.11: Robust stability analysis based on step responses

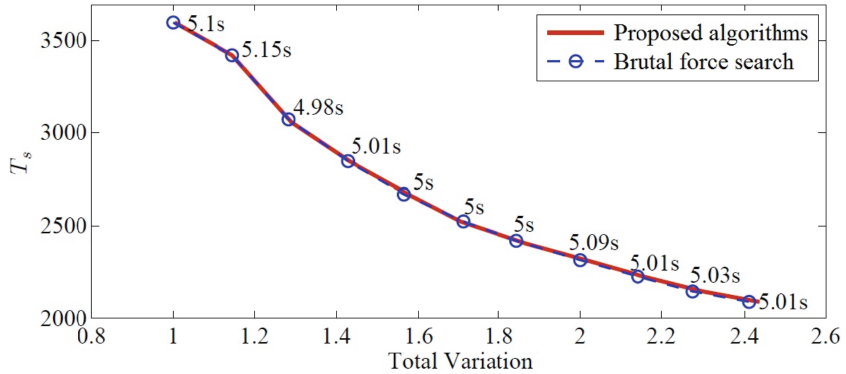


Figure 2.12: Optimality test under the uncertainty level [-40%, 100%].

of Algorithm 4 is not evaluated, as the underlying idea in finding the optimal settling time is the same as that of Algorithm 2 when the specification on the

Table 2.1: Performance comparison between Algorithm 2 and Algorithm 4 (unit of time: second)

OS*	10%	15%	20%	25%	30%	35%	40%	45%	50%
TV_2^*	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
TV_4^*	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
TV_2	1.19	1.3	1.39	1.49	1.6	1.7	1.8	1.88	2
TV_4	1.37	1.59	1.79	1.99	2.18	2.38	2.56	2.67	2.83
OS_2	0%	0%	0%	0.29%	6.87%	7.02%	12.40%	17.30%	20.73%
OS_4	0%	5.15%	13.19%	17.21%	22.01%	30.70%	38.92%	43.33%	49.97%
ts_2	3258	3045	2880	2760	2640	2535	2460	2370	2325
ts_4	2995	2640	2460	2310	2295	2100	2025	1980	1995
$ts_2 - ts_4$	263	405	420	450	345	435	435	390	330

worst-case total variation is directly given without considering the worst-case overshoot. For a given total variation, the brutal force search method examines all tuning points within a pre-defined gridded region of the $\lambda - \lambda_d$ space and computes the worst settling time for all the points. The closeness to optimality is controlled by the resolution of the grid. Fig. 2.12 shows the optimality test results under the uncertainty level [-40%, 100%], where we can observe that the proposed algorithm can capture the optimal solutions for all different cases. The computational time of the tuning algorithms is around 5 seconds on a desktop with Intel CORE-i5 and 6G memory, while the brutal force search takes about 15 minutes to complete the computation for a single specification of the worst-case total variation. Note that as the proposed algorithms are designed for a commercial control software, the computational efficiency is also important due to the fact that the site engineers may have to tune a large number of control loops within limited time and the user-friendliness of the software can be improved if the tuning algorithms run faster.

To compare the performance of Algorithms 2 and 4, the tuning results for the same overshoot specifications (the decay ratio for Algorithm 4 is chosen as 1/4 throughout this example). under the uncertainty level [-40%, 100%] are shown in Fig. 2.13, where the computation time for each point is also included. The values of the obtained settling times are presented in Table 2.1, where we use the subscript '2' to denote the performance indices of Algorithm 2, and use subscript '4' to denote the performance indices of Algorithm 4. From Fig. 2.13, Algorithm 2 requires a smaller computation time but gives a larger settling time, while Algorithm 4 runs a little slower but allows a smaller settling time.

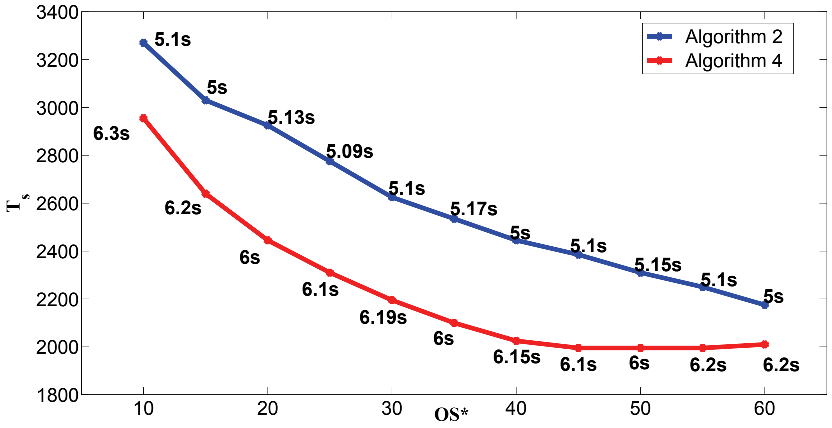


Figure 2.13: Performance comparison of Algorithms 2 and 4

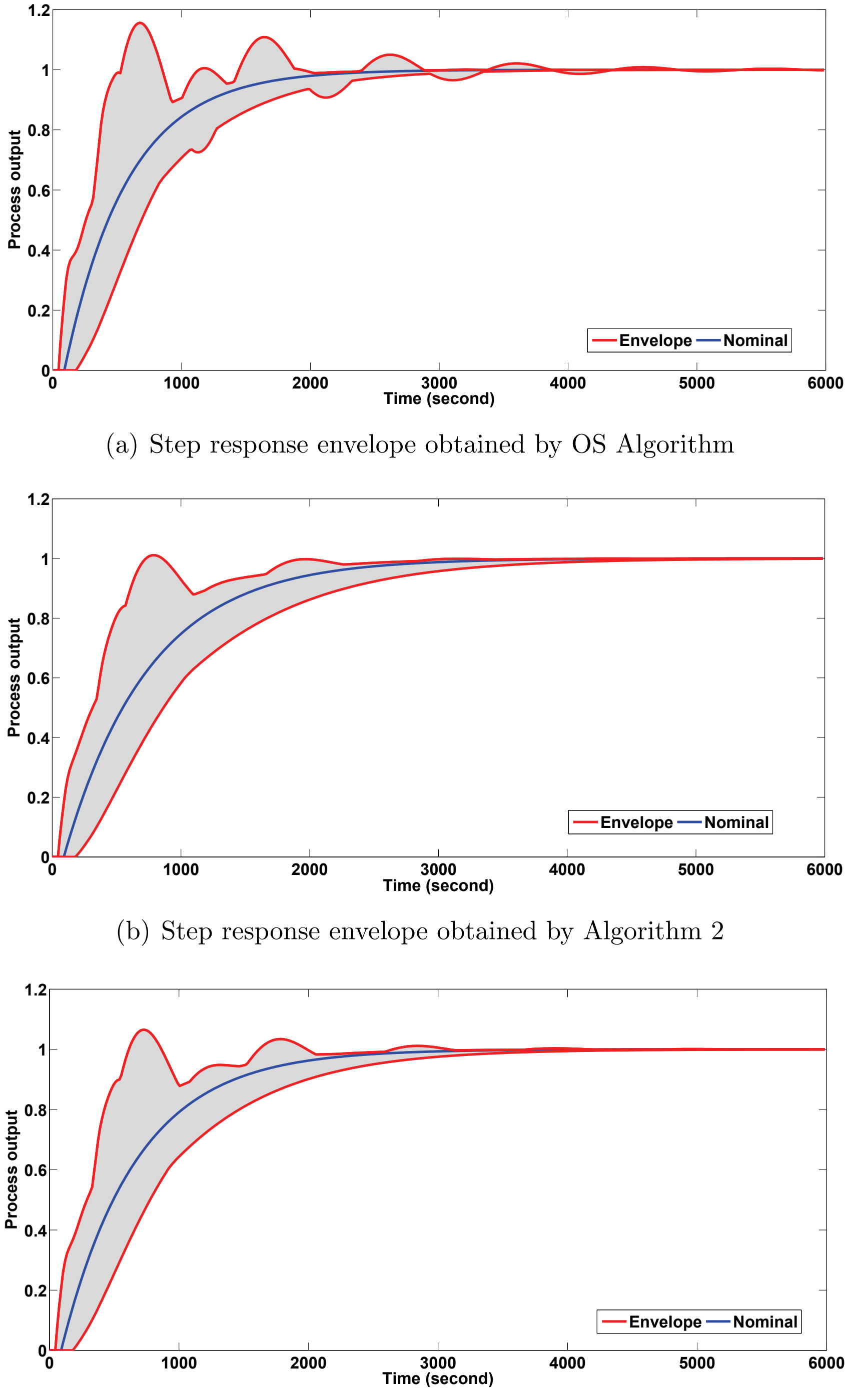
Table 2.2: Comparison of different tuning algorithms (OS$^* = 20\%$)

	TV*	TV	OS	TS (unit: second)	λ	λ_d
Algorithm 2	1.4	1.4	1.1%	2880	11.0205	5.8152
Algorithm 4	1.8	1.7675	6.5%	2520	9.6733	4.6603
OS Algorithm	–	2.54	15.6%	2310	8.1747	3.7143

The interpretation is that the Algorithm 4 uses the decay ratio to specify the total variation value for the overshoot specification, which allows a relatively larger tolerance on the worst-case total variation and thus results in relatively smaller worst-case settling times.

Now we further compare the results obtained by Algorithms 2 and 4 with the result obtained by the algorithm proposed in [53], which we call 'OS Algorithm' hereafter. Note that the OS Algorithm finds the MPC tuning parameters that minimize the worst-case settling time while considering the upper bound OS* on worst-case overshoot. The specification of worst-case overshoot is set to be OS$^* = 20\%$, and the obtained results are shown in Fig. 2.14 and Table 2.2. Based on the obtained results, the effect of introducing total variation in robust tuning is shown in two folds: a) Algorithms 2 and 4 produce relatively smoother envelope responses than the OS Algorithm, although the responses obtained by Algorithm 4 is faster than that of Algorithm 2; b) the worst-case settling times of Algorithms 2 and 4 are larger than that of OS Algorithm, which is a natural cost for the smoother responses.

Finally, we apply the tuning results to the Honeywell real time MPC +



(a) Step response envelope obtained by OS Algorithm

(b) Step response envelope obtained by Algorithm 2

(c) Step response envelope obtained by Algorithm 4

Figure 2.14: Step response envelopes of different tuning algorithms

Simulator environment. In order to consider the model mismatch, the real

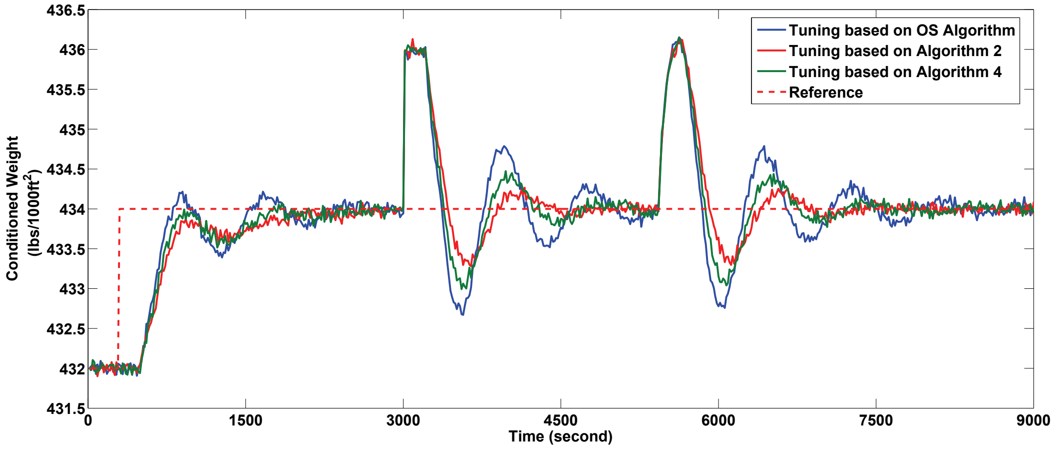


Figure 2.15: Real time MPC + simulator results of the output signal

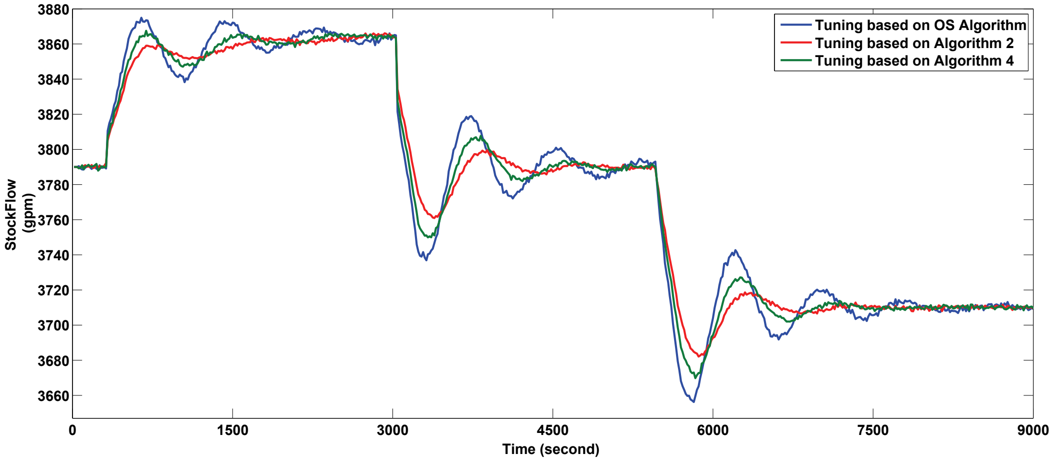


Figure 2.16: Real time MPC + simulator results of the input signal

time process is taken as

$$G_p(s) = \frac{0.0269}{119.4s + 1} e^{-179.1s}, \tag{2.39}$$

which lies within the uncertainty level [-40%,100%] of the nominal process in (2.37). The initial operating conditions are set to be $y(0) = 432$, $u(0) =$3790, according to the actual operating conditions. Then choose $\mathrm{H_p}$, $\mathrm{H_u}$, Q_1, Q_2 and Q_3, which are the optimization parameters of the MPC, to be the same as in the above tuning procedure, and use the constraints on the control signals as follows:

$$\begin{aligned} 0.9u(0) \leq U \leq 1.1u(0), \\ -0.1u(0) \leq \Delta U \leq 0.1u(0). \end{aligned} \tag{2.40}$$

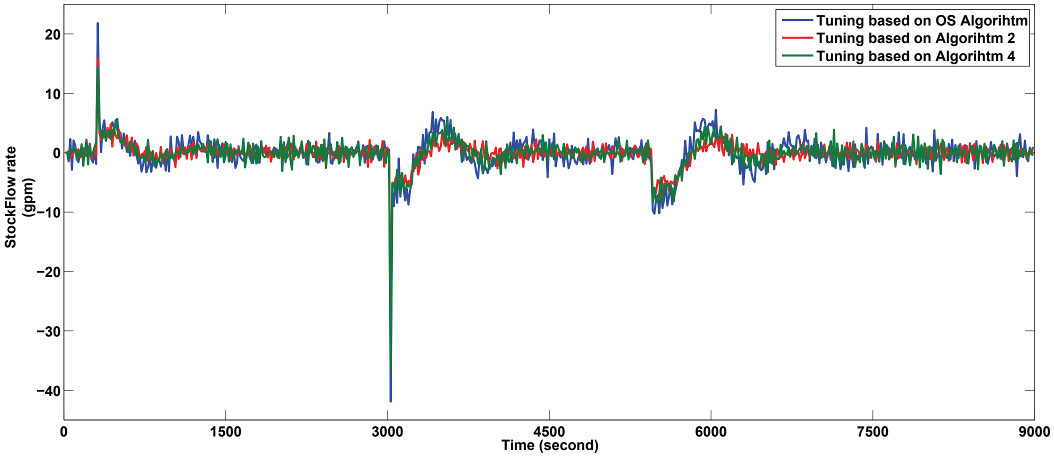


Figure 2.17: Real time MPC + simulator results of the change of the input signal

To consider the possible changes of the operating conditions, a set point change of 2 lbs/1000 ft^2 is made at $t = 300s$, also an output disturbance of 2 lbs/1000 ft^2 takes place at $t = 3000s$, an input disturbance of 80 gpm is introduced at $t = 5250s$, and the measurement noise is taken to be zero-mean Gaussian with standard deviation 0.1 lbs/1000 ft^2. The tuning results obtained in Table 2.2 are applied to the process simulator and Honeywell real-time MPC for performance comparison. The results of output signals, input signals and the changes of the input signals are shown in Figs. 2.15∼2.17, respectively. Despite the huge model uncertainty and the measurement noise, the system responses from all the three pairs of λ-values can always robustly track the reference signal for all the operating condition changes, which indicates the effectiveness of the proposed algorithms. Besides, the effect of the consideration of total variation is reflected again in Figs. 2.15 and 2.16, as the proposed algorithms yield relatively smoother output/input signal compared with that obtained without taking total variation into account (namely, the OS Algorithm). It is also worth mentioning that the maximum and minimum values in control signals and the changes of the control signals are a bit far from the limits in the constraints in (2.40), which explains why the performance corresponding to the tuning parameters obtained from the proposed methods are not affected by the constraints that much. In general, the constraints in the papermaking processes are normally loose, and therefore the proposed algorithms can guarantee the robust stability and performance, as has been shown in this example. Besides, the proposed algorithms can also be applied for the constrained case, since the empirical monotonicity and unimodality properties utilized in this work remain valid for the constrained case in general; however, a real-time MPC solver has to be utilized in Algorithms 1 and 3 to calculate the worst-case performance indices, which may significantly increase the corresponding computational time - the overall tuning time will increase from about 5 seconds to about 110 seconds (on a desktop with Intel Core i-5 and 6G RAM).

2.7 Summary

In this work, a 2-DOF MPC tuning problem is considered to minimize the worst-case settling time while considering the constraints on the worst-case overshoot and worst-case total variation. As the specification on total variation is difficult to make by the end users, two alternative methods are proposed to determine an acceptable total variation automatically, which leads to two algorithms that heuristically solve the MPC parameter tuning problem. Extensive comparison results indicate that improved performance in terms of smoother responses can be obtained by the tuning algorithms.

Chapter 3

User-Friendly Robust Tuning for MIMO MD-MPC*

3.1 Introduction

In this chapter, we focus on the controller tuning problem of machine-directional predictive control for MIMO paper-making processes with uncertain model parameters. The starting point of this work is the 2-DOF MPC structure proposed in [12]. In this framework, two filters F_r and F_d with dynamics adjusted by $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$, are added to the closed-loop system, and then we can tune the performance of the MPC controller by adjusting $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ with the original tuning weights fixed to simplify the tuning procedure. In last chapter, MD-MPC tuning problems for SISO case have been considered and solved under a similar framework. As the MD processes also have MIMO plants, an easy-to-use MIMO MD-MPC tuner is also of high demand. Thus, based on the 2-DOF MPC structure, we tune the $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ (which we refer to as λ-parameters hereafter, and are vectors with appropriate dimensions) for the desired robust performance of the closed-loop system, with the MPC penalty weights set to pre-assigned values. Based on the industrial experience, the nominal model of the real MIMO plant is available and rather than unstructured uncertainty, which is typically used to date but difficult to understand for end users, parametric uncertainties are considered in this work. The tuning objectives are specified via the overshoot, total variation and settling time for each output of the MIMO system; this enhances the easy-to-use feature of

*Parts of the results presented in this chapter are published in Control Engineering Practice, vol. 55, pp. 1-12, 2016.

the designed MIMO MD-MPC tuner, yet complicates the problem at hand. As dead time and model mismatch are unavoidable in process operation and model identification, the output responses of the closed-loop system cannot be expressed in an explicit way, and therefore the tuning problem is difficult to formulate. On the other hand, the complexity of the MIMO MD-MPC tuning problem increases with the system size, and how a specific pair of the λ-parameters (e.g., λ_i and $\lambda_{d,i}$ in vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$) affects the closed-loop responses of all the outputs of the MIMO system is unclear because of the multi-variable system structure. Besides, to further improve the easy-to-use feature of the proposed tuning algorithm, the overall time consumption of the MIMO MD-MPC tuning algorithm is not only limited by a pre-specified amount of time, but also required to be predictable without running the algorithm. In regard of the aforementioned challenges, we propose a multi-counter line based algorithm for the MIMO MD-MPC robust tuning problem. The contributions of the work are as following:

- In order to characterize the extreme step responses of each output for a set of MIMO systems described by parametric uncertainties and to calculate the worst-case time-domain measures, a fast robust performance evaluation technique is developed.

- A user-friendly MIMO MD-MPC tuning problem is constructed, and then transformed into a number of individual MISO tuning problems, based on which the tuning problem is simplified. Based on the robust performance evaluation method and the simplified tuning problem, a fast multi-variable tuning method is developed for the MIMO MD-MPC, based on which the controller parameters can be tuned for satisfactory performance within acceptable computation time.

- An efficient technique to predict the overall computation time of the proposed MIMO MD-MPC tuning algorithm is proposed, based on which the end users can predict the tuning time without running the algorithm.

3.2 Preliminaries and Problem Formulation

In what follows, the 2-DOF MIMO MD-MPC framework developed in [12] is introduced; the user-friendly tuning objective is also proposed in this section.

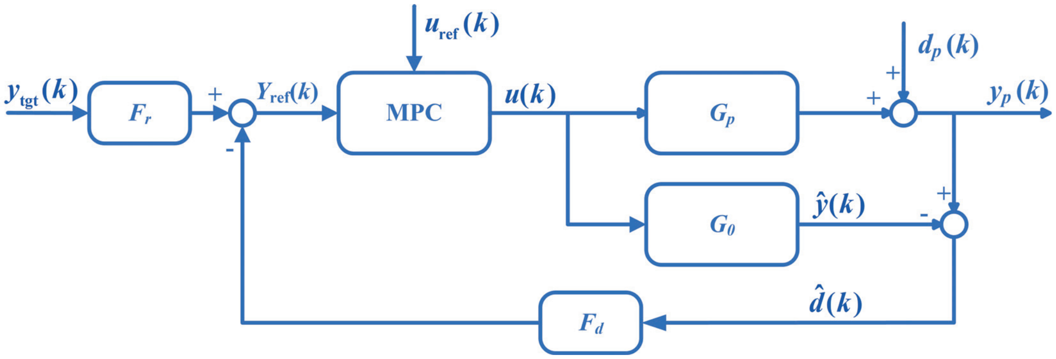


Figure 3.1: The MIMO 2-DOF MPC structure

3.2.1 Nominal system and uncertainty

In Fig. 3.1, $G_p \in \mathbb{R}^{m\times n}$ and $G_0 \in \mathbb{R}^{m\times n}$ are the transfer function matrices of the real process and nominal model respectively. The FOPDT structure is utilized to model the subsystems in G_p and G_0, as it can provide a good approximation of the process [13] and is also easy to understand by the operators of paper machines. Therefore the real process G_p can be represented as

$$G_p(s) = \begin{bmatrix} G_p^{11}(s) & \cdots & G_p^{1n}(s) \\ \vdots & \ddots & \vdots \\ G_p^{m1}(s) & \cdots & G_p^{mn}(s) \end{bmatrix},$$
$$G_p^{ij}(s) = \frac{k_{ij}}{\tau_{ij}s+1}e^{-T_{ij}s}, 1 \le i \le m, 1 \le j \le n, \tag{3.1}$$

where k_{ij}, τ_{ij} and T_{ij} are the real process gain, time constant and dead time for G_p^{ij}. The discretized model of G_p^{ij} is $G_p^{ij}(z) = k_{ij}\frac{b_{ij}z^{-1}}{1-a_{ij}z^{-1}}z^{-T_{ij}^d}$, where $a_{ij} = e^{-\Delta T/\tau_{ij}}, b_{ij} = 1 - a_{ij}$, ΔT indicates the sampling time, and T_{ij}^d is the discretized time delay. As model mismatch, disturbance and noises exist in practice, $G_p(s)$ cannot be known accurately. Thus, a nominal model $G_0(s)$ is identified as the approximation of $G_p(s)$:

$$G_0(s) = \begin{bmatrix} G_0^{11}(s) & \cdots & G_0^{1n}(s) \\ \vdots & \ddots & \vdots \\ G_0^{m1}(s) & \cdots & G_0^{mn}(s) \end{bmatrix},$$

$$G_0^{ij} = \frac{k_{ij}^0}{\tau_{ij}^0 s + 1} e^{-T_{ij}^0 s}, 1 \leq i \leq m, 1 \leq j \leq n, \tag{3.2}$$

where k_{ij}^0, τ_{ij}^0 and T_{ij}^0 are usually obtained from some commercial control software based on the input/output data of the real plant, and the discretization of model is achieved similarly as that of the process model. However, the difference between the real process and nominal model is inevitable so that the uncertainties in the model parameters must be considered. Two different kinds of uncertainties are normally used in the robust control, namely, parametric uncertainty and unstructured uncertainty. As the unstructured uncertainty is not familiar to our end users, parametric uncertainty is considered for MD-MPC parameter tuning, which can be denoted in the following form

$$\begin{aligned} & k_{ij} \in [\underline{k}_{ij},\ \overline{k}_{ij}],\ \tau_{ij} \in [\underline{\tau}_{ij},\ \overline{\tau}_{ij}],\ T_{ij} \in [\underline{T}_{ij},\ \overline{T}_{ij}], \\ & 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{3.3}$$

where $\underline{k}_{ij}$, $\overline{k}_{ij}$, $\underline{\tau}_{ij}$, $\overline{\tau}_{ij}$ and $\underline{T}_{ij}$, $\overline{T}_{ij}$ are the lower and upper bounds of the mismatch on the model parameters. Thus, we can represent the uncertain models as

$$\begin{aligned} \Pi := \{G_p(s) :& k_{ij} \in [\underline{k}_{ij},\ \overline{k}_{ij}],\ \tau_{ij} \in [\underline{\tau}_{ij},\ \overline{\tau}_{ij}], \\ & T_{ij} \in [\underline{T}_{ij},\ \overline{T}_{ij}], 1 \leq i \leq m, 1 \leq j \leq n\}. \end{aligned} \tag{3.4}$$

Note that given the multivariate structure of the system, it is difficult to find an analytical connection between the parametric uncertainty and the unstructured uncertainty. However, the proposed tuning algorithms in this chapter provide an alternative way to directly handle the parametric uncertainty so that the robust stability and performance can be guaranteed.

3.2.2 MPC controller

First define H_p, H_u as the prediction and control horizon, respectively. The MPC in Fig. 3.1 amounts to solving the following finite-horizon optimal control problem [12]

$$\begin{aligned}
&\min_{\Delta U}\ J = (E_y(k))^T Q_1 (E_y(k)) + \Delta U(k)^T Q_2 \Delta U(k) \\
&+ (E_u(k))^T Q_3 (E_u(k)) \\
&\text{s.t.}\ \ \hat{x}(k+i) = A^i \hat{x}(k) + \sum_{j=1}^{\min\{H_u,i\}} A^{i-j} B \Delta u(k+j-1), \\
&\hat{y}(k+i) = C\hat{x}(k+i),\ \text{for}\ i = 1, 2, \ldots, H_p,
\end{aligned} \tag{3.5}$$

where

$$\begin{aligned}
E_y(k) &= \hat{Y}(k) - Y_{\text{ref}}(k) \\
&= \begin{bmatrix} \hat{y}(k+1) - y_{\text{ref}}(k+1) \\ \hat{y}(k+2) - y_{\text{ref}}(k+2) \\ \vdots \\ \hat{y}(k+H_p) - y_{\text{ref}}(k+H_p) \end{bmatrix},
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
E_u(k) &= U_{H_u}(k) - U_{\text{ref}}(k) \\
&= \begin{bmatrix} u(k) - u_{\text{ref}}(k) \\ u(k+1) - u_{\text{ref}}(k+1) \\ \vdots \\ u(k+H_u-1) - u_{\text{ref}}(k+H_u-1) \end{bmatrix},
\end{aligned} \tag{3.7}$$

$$\Delta U(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+H_u-1) \end{bmatrix}, \tag{3.8}$$

$$U_{H_u}(k) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k-1) + \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \cdots & 1 & 1 \end{bmatrix} \Delta U(k). \tag{3.9}$$

U_{ref} and Y_{ref} are the reference signal vector of U_{H_u} and $\hat{Y}$, respectively, and Q_1, Q_2 and Q_3 are penalty weights.

Then, we can represent the MPC as a quadratic programming problem with the following objective

$$J = \Delta U^T(k) H \Delta U(k) + 2\varphi^T \Delta U(k), \tag{3.10}$$

where H and φ are the Hessian matrix and gradient matrix, respectively. Therefore, the analytical expression of $\Delta U(k)$ that minimizes (3.10) is

$$\begin{aligned}\Delta U(k) =& -H^{-1}\varphi \\ =& K_x \hat{x}(k) + K_{u-1} u(k-1) + K_{yt} Y_{\text{ref}} + K_{xt} U_{\text{ref}},\end{aligned} \tag{3.11}$$

where $K_x, K_{u-1}, K_{yt}, K_{xt}$ are matrices related to the parameters in MPC and the nominal model (see [52] for the detailed expressions of the matrices).

Besides, compared with the traditional MPC, an additional penalty term E_u is considered in the cost function for the economic purpose. As the energy consumption is an important element affecting total profits of a paper mill, E_u can be utilized to minimize the operation cost of paper machines. Note that if there are more manipulated variables than controlled variables plus (active) constraints, degrees of freedom to execute steady-state optimization exist for the system [13]. Therefore, for some paper-making systems, the penalty on the difference between the actual input and the corresponding ideal steady state value u_{ref} can be utilized in the MPC to reduce the cost without downgrading the quality of the product.

Remark 3.1. *In paper-making processes, there are only constraints on the input signal U_{H_u} and the change of the input signal ΔU. Based on industrial experience, these constraints are relatively loose compared with actual signals appeared in the control loop, and therefore the constraints are not considered in the tuning problem to save the computational resource for user-friendliness. Besides, test results on industrial example with the actual constrained MPC show that the performance of the algorithm is not evidently influenced by the actual constraints (see Section 3.5 for the detailed results and discussions).*

3.2.3 2-DOF tuning

Given the MPC controller and system model in previous subsections, the filters F_r and F_d, as shown in Fig. 3.1, are designed to filter the target of the output ($y_{\text{tgt}}(k)$) and the estimated disturbance ($\hat{d}(k) \triangleq y_p(k) - \hat{y}(k)$), which constitute the MIMO 2-DOF MPC structure. Based on the filtered signals, the reference trajectory is obtained as follows:

$$Y_{\mathrm{ref}}(k) = \begin{bmatrix} y_{\mathrm{ref}}(k+1) \\ y_{\mathrm{ref}}(k+2) \\ \vdots \\ y_{\mathrm{ref}}(k+H_p) \end{bmatrix} = F_r(y_{\mathrm{tgt}}(k)) - F_d(\hat{d}(k)), \tag{3.12}$$

where $F_r(\cdot)$ and $F_d(\cdot)$ are projection filters to implement $f_r(z)$ and $f_d(z)$, based on $y_{\mathrm{ref}}(z) = f_r(z)y_{\mathrm{tgt}}(z) - f_d(z)\hat{d}(z)$ [12]; $f_r(z)$ and $f_d(z)$ are the reference tracking filter and disturbance rejection filter, respectively; The user-specified reference tracking filter $f_r(z)$ is taken to have the form

$$f_r(z) = \begin{bmatrix} f_{r,1}(z) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & f_{r,m}(z) \end{bmatrix}, \text{ and}$$

$$f_{r,i}(z) := \frac{b_{r,i}z^{-1}}{1-a_{r,i}z^{-1}}z^{-T_i}, \; 1 \le i \le m, \tag{3.13}$$

where $a_{r,i} = e^{-\frac{Ts}{\tau_i \lambda_i}}$, $b_{r,i} = 1 - a_{r,i}$; $f_{r,i}(z)$ is the reference tracking filter for the ith output; we define $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m]$ as the reference tracking ratio vector. Similarly, $f_d(z)$ has the following form

$$f_d(z) = \begin{bmatrix} f_{d,1}(z) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & f_{d,m}(z) \end{bmatrix}, \text{ and}$$

$$f_{d,i}(z) := \frac{b_{d,i}z^{-1}}{1-a_{d,i}z^{-1}}z^{-T_i}, \; 1 \le i \le m, \tag{3.14}$$

where $a_{d,i} = e^{-\frac{Ts}{\tau_i \lambda_{d,i}}}$, $b_{d,i} = 1 - a_{d,i}$, and $f_{d,i}(z)$ is the disturbance rejection filter for the ith output; we define $\boldsymbol{\lambda_d} = [\lambda_{d,1}, \cdots, \lambda_{d,m}]$ as the disturbance rejection ratio vector. In this way, the performance of the MPC can be tuned by $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ with fixed penalty matrices ($Q_1 = I, \; Q_2 = 0.01 \times I, \; Q_3 = 0$ in this work); and hence the MPC tuning problem is simplified [12]. It is worth noting that the function of the original penalty matrices Q_2, Q_3 can be achieved by $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$. According to the 2-DOF system structure, abrupt changes in the output reference Y_{ref} can be removed by the tuning of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ to reduce the aggressiveness of the corresponding control inputs u, as $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ affect the smoothness of the references for each output. Based on the structure of f_r and f_d shown in (3.13)-(3.14), the time constants of the filters increase with the increase of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$, and therefore larger values of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ make the reference Y_{ref} smoother, and the resultant the closed-loop system responses more sluggish. On the contrary, smaller values of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ make the system responses more aggressive. In particular, as f_d filter appears in the feedback channel, $\boldsymbol{\lambda_d}$ affects the robust stability of the closed-loop system; a larger $\boldsymbol{\lambda_d}$ improves the robustness but also increases the settling time. f_r outside the feedback loop only affects the speed of the reference signal, and therefore the increase of $\boldsymbol{\lambda}$ can reduce the values of overshoot and total variation.

In addition, the reason of utilizing the diagonal structure for the filters f_r and f_d is to make the tuning problem easier to save more computational resource. As f_r and f_d are employed to smooth the reference and estimated error trajectories for each output, the intuition of adopting the diagonal structure is to filter each trajectory only based on its original signal; besides, the diagonal form is a concise way to filter the trajectories, and is sufficient to affect the closed-loop system responses. Note that a non-diagonal structure can also be utilized to design better filters, but additional tuning parameters appear, making the tuning problem more complex and the automated tuning procedure slower. As the user-friendliness is of high priority in this work, we have selected the diagonal form for filter design to make the algorithm more computationally efficient.

Note that a key part of the 2-DOF MPC tuning problem is the design of $f_{r,i}(z)$ and $f_{d,i}(z)$ in (3.13) and (3.14). The principle to design $f_{r,i}(z)$ is that given the ith output, the open-loop transfer function from the input that dominates the output is selected to construct this filter, i.e., we select τ_i of $f_{r,i}(z)$ to be the open-loop time constant of the dominant subsystem[1] for ith output; the underlying reason is to make the speed of the closed-loop response dependent on the speed of the dominant or primary open-loop response of the system according to the requirements on a specific paper product. As for $f_{d,i}(z)$, the same design procedure is applied except that $\lambda_{d,i}$ is used instead of λ_i such that we can filter the output target and the estimated disturbance

[1] The dominant subsystem is either selected based on the physical property of the paper machine or to be the subsystem with the largest time constant in our work.

separately.

3.2.4 User-friendly tuning objectives

For the MIMO system considered in this work, the overshoot, total variation, and settling time of each output are considered as the tuning measurements. Since the parametric uncertainty (defined in (3.3)) results in a set of perturbed systems, the worst-case time domain performance indices are employed. In the following, the definitions for the worst-case overshoot, settling time and worst-case total variation are first recalled.

Definition 3.1. (**Worst-case overshoot**). *The worst-case overshoot* OS *of a set of step responses with the same final value is the maximum value in all responses minus the final value divided by the final value.*

Definition 3.2. (**Worst-case settling time**). *The worst-case settling time* $\mathrm{T_s}$ *of a set of step responses with the same final value is the minimum time required for all the responses to reach and stay within a range of pre-specified percentage of the final value.*

Definition 3.3. (**Worst-case total variation**). *Assume the system converges to the target value within n steps, the worst-case total variation* TV *is*

$$\mathrm{TV} = \max_{\mathrm{G_p} \in \Pi} \sum_{k=1}^{n} |y(k) - y(k-1)|. \tag{3.15}$$

Note that Definitions 3.1-3.3 are introduced for characterizing the worst-case performance for a set of step responses for SISO systems, and thus are applicable for each of the outputs of the MIMO system. In this work, we allow the end users to specify their requirements on each output directly via the worst-case time domain performance indices. This improves the user-friendless of the proposed MIMO MPC tuner. In addition, the tuning time, defined as t_λ, is also an important factor in a successful industrial MPC tuner design. According to the guideline of commercial paper machine tuner design, the total time consumption of the tuning algorithm should not be more than half a minute; and this requirement needs to be considered also as a hard constraint for the problem at hand. Note that the requirement on t_λ is selected according to the size of the real MD process (e.g., no large than 7 outputs and

8 inputs), and the average time that a site engineer can spend on the controller tuning procedure based on industrial experience.

Remark 3.2. *Note that the tuning time was normally not considered in the existing MIMO MPC tuning algorithms, because 1) a limited computation time might affect feasibility of these algorithms and 2) the computation time was normally not predictable. A hard constraint on the tuning time would further increase the difficulty of solving the MIMO MPC tuning problem.*

Based on the factors mentioned above, the tuning target is twofold: 1) the robust stability of the MIMO system is achieved; 2) the closed-loop responses of all the outputs track the reference signals with small OS, $\mathrm{T_s}$ and TV, given the parametric uncertainty defined in (3.3). Besides, the $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ should also be tuned within the acceptable tuning time. To do this, we formulate the tuning problem as

$$\begin{aligned}&\min_{\boldsymbol{\lambda},\boldsymbol{\lambda_d}} ||\mathbf{T}_s(\boldsymbol{\lambda},\boldsymbol{\lambda_d})||_\infty,\\ &\text{s.t. } \mathrm{OS}_i(\boldsymbol{\lambda},\boldsymbol{\lambda_d}) \le \mathrm{OS}_i^*,\ 1 \le i \le m,\\ &\quad\ \ \mathrm{TV}_i(\boldsymbol{\lambda},\boldsymbol{\lambda_d}) \le \mathrm{TV}_i^*\ 1 \le i \le m,\\ &\quad\ \ t_\lambda \le 30s, \end{aligned} \tag{3.16}$$

where $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_m]$, $\boldsymbol{\lambda_d} = [\lambda_{d,1}, \dots, \lambda_{d,m}]$. Note $\mathbf{OS}(\boldsymbol{\lambda},\boldsymbol{\lambda_d}) := [\mathrm{OS}_1(\boldsymbol{\lambda},\boldsymbol{\lambda_d}), \mathrm{OS}_2(\boldsymbol{\lambda},\boldsymbol{\lambda_d}), \dots, \mathrm{OS}_m(\boldsymbol{\lambda},\boldsymbol{\lambda_d})]$, $\mathbf{TV}(\boldsymbol{\lambda},\boldsymbol{\lambda_d}) := [\mathrm{TV}_1(\boldsymbol{\lambda},\boldsymbol{\lambda_d}),\ \mathrm{TV}_2(\boldsymbol{\lambda},\boldsymbol{\lambda_d}), \dots, \mathrm{TV}_m(\boldsymbol{\lambda},\boldsymbol{\lambda_d})]$ and $\mathbf{T}_s(\boldsymbol{\lambda},\boldsymbol{\lambda_d}) := [\mathrm{T}_{s,1}(\boldsymbol{\lambda},\boldsymbol{\lambda_d}),\ \mathrm{T}_{s,2}(\boldsymbol{\lambda},\boldsymbol{\lambda_d}), \dots, \mathrm{T}_{s,m}(\boldsymbol{\lambda},\boldsymbol{\lambda_d})]$ denote the worst-case overshoots, total variations and settling times given the $(\boldsymbol{\lambda},\boldsymbol{\lambda_d})$, for all the outputs of the MIMO system $G_p(s)$ within parametric uncertainty region, i.e., $G_p(s) \in \Pi$. $\mathbf{OS}^* := [\mathrm{OS}_1^*,\ \mathrm{OS}_2^*, \dots, \mathrm{OS}_m^*]$, $\mathbf{TV}^* := [\mathrm{TV}_1^*,\ \mathrm{TV}_2^*, \dots, \mathrm{TV}_m^*]$ are the specifications on the overshoots and total variations. Note that in (3.16), we tune $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ to find the smallest worst-case settling times of all the outputs while require the corresponding worst-case OS and TV to be no greater than the pre-determined specifications, and require the tuning time is no more than half a minute, i.e., $t_\lambda \le 30s$.

It is worth noting that since the explicit formulas for the performance measures employed in (3.16) are generally not available, the constraints and cost of the optimization problem become implicit, and therefore a typical option is to adopt evolution based optimization techniques (e.g., the genetic algorithm), which, however, increase the computational complexity significantly so that the constraint on tuning time can not be satisfied. The proposed algorithm utilizes the empirical (monotone/unimodal) relationship between the tuning parameters and performance measures obtained based on the industrial experience and extensive simulations, and then seeks for the tuning parameters via the line search methods. Although the optimality can not be guaranteed theoretically, satisfactory results can normally be achieved, as shown in the example section. Besides, the line search structure of the proposed algorithm also allows us to develop the tuning time prediction technique, which allows the users to know the tuning time beforehand.

Besides, the optimization problem in (3.16) is always feasible for MD paper-making processes; the reason is as follows: 1) the first two constraints in (3.16) can be satisfied when $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ are selected large enough, because the TV approaches 1 and OS approaches 0 as $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ increase. In the proposed algorithm, an adequately large upper limit is chosen for both $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$, which guarantees the feasibility of these two constraints; 2) as the size of MD process is limited (normally no larger than 7 outputs and 8 inputs), the feasibility of the constraint on tuning time is verified by testing the proposed algorithm on the largest MD process.

3.3 Envelope Algorithm: the MIMO Case

3.3.1 Proposed performance evaluation technique

In this section, we propose a performance evaluation method to compute the worst-case OS, $\mathrm{T_s}$ and TV for all the outputs of the MIMO system, which is an essential part to handle (3.16). For a specific pair of $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$, the goal here is to calculate the envelopes that can serve as the upper and lower bounds of all the possible step responses for all uncertain MIMO systems satisfying $G_p(s) \in \Pi$. As the system considered has multiple outputs, the envelopes are represented for each of the outputs, based on which the outcome of selecting a specific pair of $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$ can be figured out directly.

In [53], a visualization method was developed to obtain the envelope responses for a group of SISO FOPDT systems ($G_p^*(s)$) under parametric uncertainty:

$$G_p^*(s) \in \Pi_e := \{G_p^*(s) : k \in [\underline{k},\ \overline{k}], \tau \in [\underline{\tau},\ \overline{\tau}], T \in [\underline{T}, \overline{T}]\},$$

via 8 extreme-case systems:

$$\Pi_E := \{G_p^*(s) : k \in \{\underline{k},\ \overline{k}\}, \tau \in \{\underline{\tau},\ \overline{\tau}\}, T \in \{\underline{T}, \overline{T}\}\},$$

where $k,\ \tau,\ T$ are the gain, time constant and time delay respectively. Note that as the visualization problem can be viewed as a constrained optimization problem, the optimal solution can be achieved via examining every possible combination of active constraints based on the Karush-Kuhn-Tucker condition. Due to the requirement on the tuning time, enumerating every combination of active constraints (including the case that no constraint is active, namely, the so-called 'interior' case) is computationally impossible; therefore, a suboptimal solution in the sense that only a subset of active constraint combinations focusing on the extreme cases is utilized in our work. Although this suboptimal solution is not always guaranteed to be optimal, extensive simulation results indicate that the technique utilized is effective, as industrial experience suggests that worst-case step-response behavior is normally experienced by extreme process parameters. Although all subsystems in the MIMO process $G_p(s)$ have the FOPDT structure, and the parametric uncertainty of $G_p(s)$ is constructed by considering all subsystems in a way like Π_e, the approach mentioned above cannot be used directly to the MIMO process due to the following factors: 1) the number of extreme-case systems from [53] depends on the number of model parameters, and becomes a large number even for a low dimensional MIMO system (e.g., a 2-by-2 system consists of 4 FOPDT subsystems, and thus requires 8^4 extreme-case systems based on the approach in [53]); 2) the computation time of the visualization method in [53] increases rapidly as the number of extreme-case systems increases; 3) as the tuning procedure should cost less than 30 seconds, and the worst-case performance will be evaluated many times in the tuning algorithm, the performance evaluation algorithm should run very fast.

To deal with this problem, we first define the parameters of the MIMO system considered as

$$\boldsymbol{k} = \begin{bmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mn} \end{bmatrix}, \ \boldsymbol{\tau} = \begin{bmatrix} \tau_{11} & \cdots & \tau_{1n} \\ \vdots & \ddots & \vdots \\ \tau_{m1} & \cdots & \tau_{mn} \end{bmatrix}$$

$$\boldsymbol{T} = \begin{bmatrix} T_{11} & \cdots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{m1} & \cdots & T_{mn} \end{bmatrix}. \tag{3.17}$$

Then, following the idea of characterizing a set of uncertain systems via a polytopic system (specified by extreme uncertain systems on the vertices) in multi-variable robust control theory [2, 62], the MIMO system with parametric uncertainty defined in (3.3) can be approximated as

$$\begin{aligned} \mathcal{P}_E := \{G_p(s) :& \boldsymbol{k} \in \{\underline{\boldsymbol{k}},\ \overline{\boldsymbol{k}}\}, \boldsymbol{\tau} \in \{\underline{\boldsymbol{\tau}},\ \overline{\boldsymbol{\tau}}\}, \\ & \boldsymbol{T} \in \{\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}\}\}, \end{aligned} \tag{3.18}$$

where $\underline{\boldsymbol{k}}$, $\underline{\boldsymbol{\tau}}$ and $\underline{\boldsymbol{T}}$ denote all entries in $\boldsymbol{k}$, $\boldsymbol{\tau}$, and $\boldsymbol{T}$ taking their minimum values in the uncertainty region, and $\overline{\boldsymbol{k}}$, $\overline{\boldsymbol{\tau}}$ and $\overline{\boldsymbol{T}}$ correspond to all elements taking their maximum. Note that $\mathcal{P}_E$ only consists of eight extreme-case systems. Therefore, the envelopes of the MIMO system with parametric uncertainty defined in (3.3) can be obtained via $\mathcal{P}_E$ with significantly reduced computation time.

Based on $\mathcal{P}_E$, the envelope responses of each output of the MIMO system with MD-MPC can be obtained. The detailed procedure is summarized in Algorithm 5.

Algorithm 5 MIMO envelope algorithm

```
Input $\boldsymbol{\lambda_d}$, $\boldsymbol{\lambda}$ and the uncertainty intervals $[\underline{\boldsymbol{k}}, \overline{\boldsymbol{k}}]$, $[\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}]$ and $[\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}]$;
Determine the eight worst-case uncertain systems in $\mathcal{P}_E$ based on extreme combinations of the uncertainty intervals, and then calculate the output responses $\mathrm{Y}_i = [y_1, \ldots, y_m]_i^{\mathrm{T}}$ for the $i$th extreme system $(i = 1, 2, \ldots, 7, 8)$ in $\mathcal{P}_E$ based on the analytical expression of $\Delta U$ in (3.11).
Obtain the upper envelope for all the outputs $\{\overline{y}_1, \ldots, \overline{y}_m\}$ by calculating $\max_{i \in \{1,2,\ldots,7,8\}} \mathrm{Y}_i(k,:)$, $k = 1, 2, \ldots, m$.
Obtain the lower envelope for all the outputs $\{\underline{y}_1, \ldots, \underline{y}_m\}$ by calculating $\min_{i \in \{1,2,\ldots,7,8\}} \mathrm{Y}_i(k,:)$, $k = 1, 2, \ldots, m$.
```

Specifically, Algorithm 5 is mainly composed of the following steps: 1) construct 8 extreme case uncertain systems based (3.17)-(3.18); 2) run closed-loop simulation for each of the 8 extreme systems with the MPC introduced in Section 3.2, and obtain the output vector $Y_i, i = 1, \ldots, 8$ for the 8 extreme systems; here, the simulation period is selected as $12(\max_i(\max_j(\overline{T}_{ij}))+$

$\max_i(\max_j(\overline{\tau}_{ij})))$, where $\overline{T}_{ij}$ and $\overline{\tau}_{ij}$ indicate the (i, j)-th element of the extreme model parameter matrices $\overline{T}$ and $\overline{\tau}$ for time constant and delay in (3.17)-(3.18); 3) to obtain the upper envelope of the kth output, take the maximum value of the kth output in all Y_i at each the time step; 4) take the minimum value of the kth output in all Y_i at each time step to construct the lower envelope.

Extensive simulation results indicate that the worst-case OS, $\mathrm{T_s}$ and TV can be approximated based on the responses of the extreme systems, for which certain combinations of the critical/extreme uncertain model parameters are reached simultaneously. The underlying theoretic support is that the performance of a set of uncertain systems described by a polytopic system can be characterized only by the set of systems with vertex parameters of the polytopic uncertainty [2, 62] and in particular, can be captured only by the set of the worst-case vertex systems for the purpose of computation cost reduction [50]; thus, the proposed performance evaluation approach is able to characterize the time domain measures employed in the controller tuning. In addition, the actual performance of the proposed method is also verified by simulations with different processes extensively in next subsection.

3.3.2 Extensive simulation study of the proposed performance evaluation technique

In order to further evaluate the performance of Algorithm 5 in terms of accurate calculation of the envelope responses, the extensive simulation-based method is employed. For all different types of MIMO processes considered in the real paper-making machine directional control (the largest MIMO process used in the MD-MPC is about 7 inputs and 8 outputs), the following test procedure is utilized: we set λ-parameters to different values, and compare the envelopes obtained from Algorithm 5 with the step responses from a large number ($\geqslant$500) of uncertain systems, the parameters of which are randomly selected based on the interval uncertainties, to accomplish the verification. The obtained results show that Algorithm 5 works effectively in all the MIMO MD processes of the paper machine. Due to the space limitation, here we only show the results (see Figure 3.3) from one process, which is a 2-by-3 MD process of a paper machine, controlling the Dry weight and Moisture of the paper products by Stockflow, Steam and Speed. An illustration plot of such process is shown in Figure 3.2. The readers may also refer to Figs. 1 to

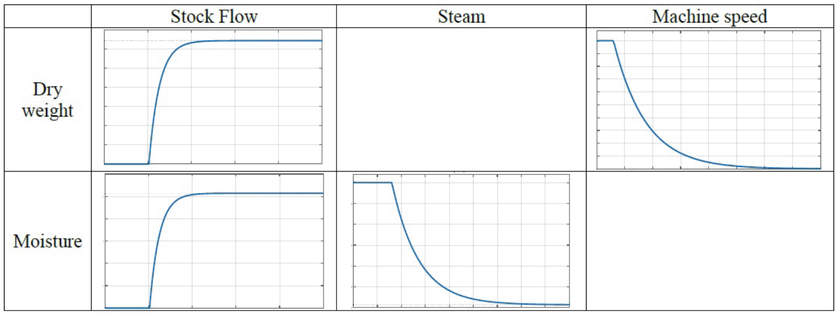


Figure 3.2: Illustration of the typical 2-by-3 MD process

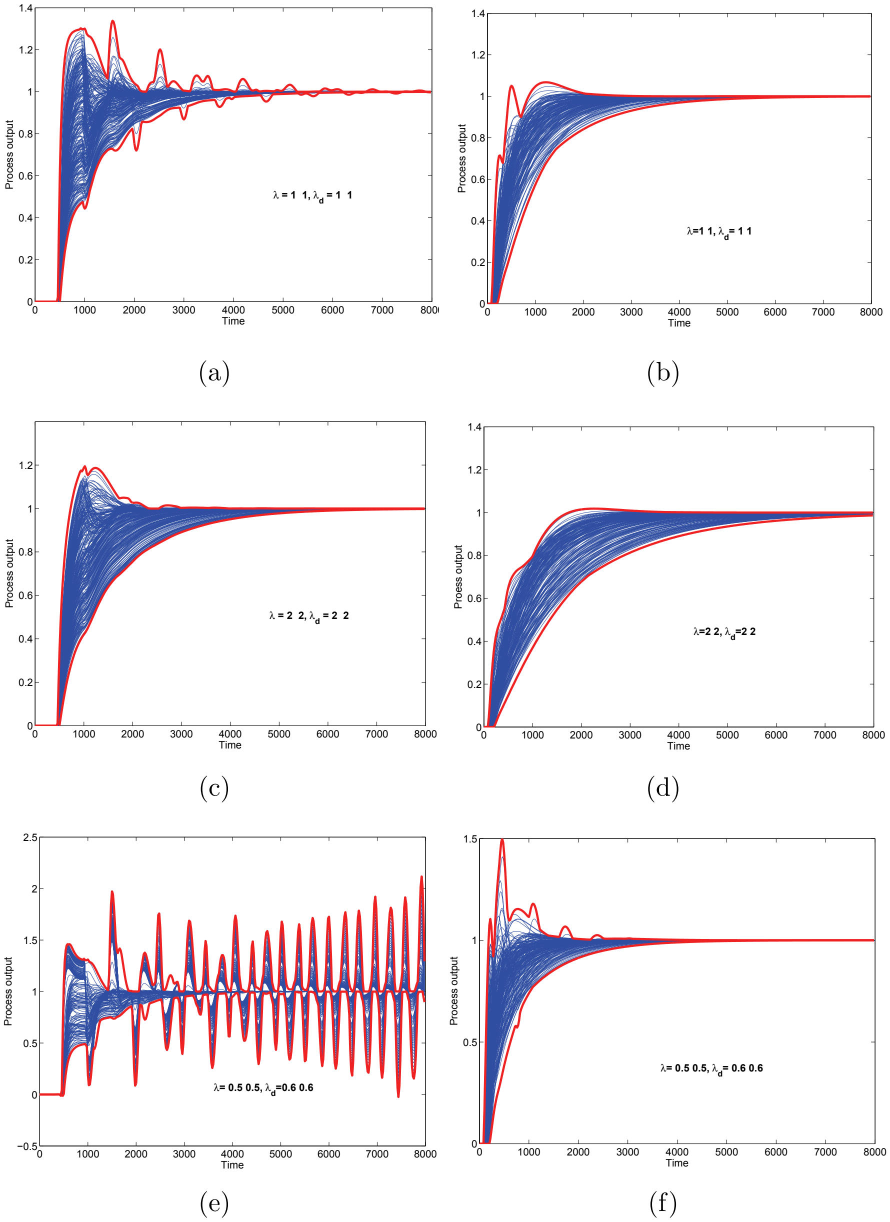
4 and the corresponding descriptions in [13] for a more detailed illustration of real paper-making processes. Note that in Figure 3.3 we show two typical situations that may happen during the tuning: 1) the system is stable, see Fig. 3.3 (a)-(b) and Fig. 3.3 (c)-(d); 2) the system is unstable, see Fig.3.3 (e) and (f). The values of $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$ are shown in the figures, and the uncertainty level is considered as $[-30\%,\ 50\%]$. Here the uncertainty level $[-\underline{r},\ \overline{r}]$ denotes the intervals of the model parameters based on their nominal values:

$$
\begin{aligned}
k_{ij} &\in [(1-\underline{r})k_{ij}^0,\ (1+\overline{r})k_{ij}^0],\\
\tau_{ij} &\in [(1-\underline{r})\tau_{ij}^0,\ (1+\overline{r})\tau_{ij}^0],\\
T_{ij} &\in [(1-\underline{r})T_{ij}^0,\ (1+\overline{r})T_{ij}^0].
\end{aligned}
$$

In Figure 3.3, we can see that for each output the corresponding envelopes all serve as tight upper and lower bounds for all step responses from different MIMO systems whose parameters satisfying the user-determined parametric uncertainty; the resultant OS, T$_s$ and TV can then be calculated directly from the envelopes. Meanwhile, the proposed method is also computational efficient, because implementing this technique for the largest MIMO process (7 outputs and 8 inputs, which we refer to 7-by-8 hereafter) accounts for only 0.47 seconds on a computer with Intel i5 CPU and 6 GB RAM.

3.4 MIMO Tuning Algorithm

The MIMO tuning problem is first simplified in this section, and then a fast multi-variable MPC tuning algorithm is designed to efficiently tune the ratio vector $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$ for desired time-domain performance.



(a) (b)

(c) (d)

(e) (f)

Figure 3.3: Numerical verification of the proposed performance visualization technique

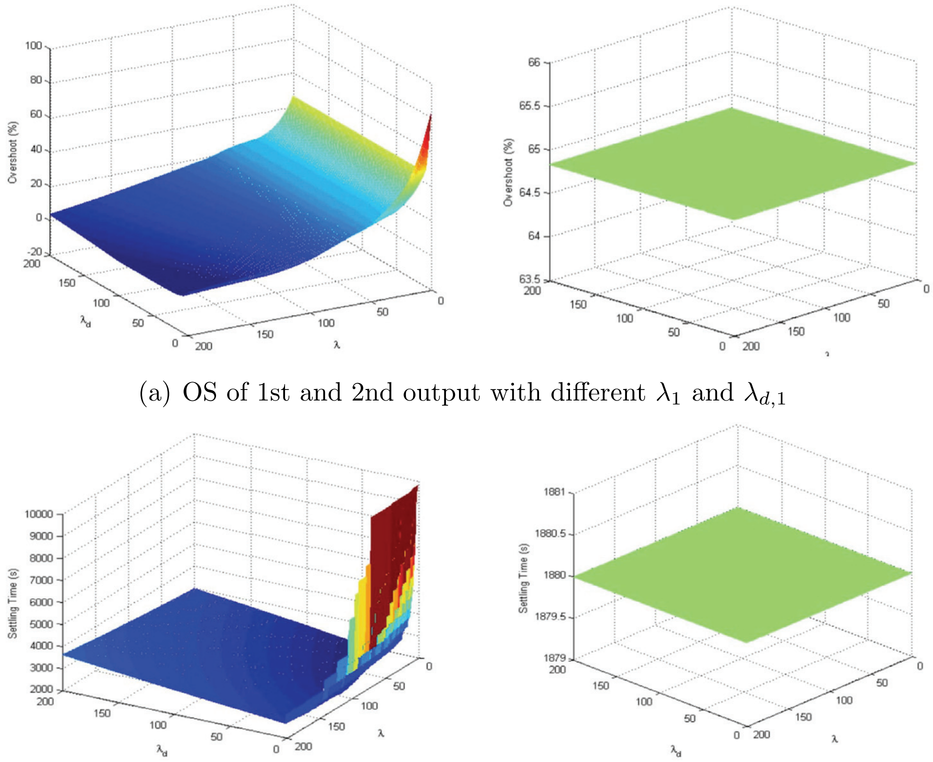
3.4.1 MIMO tuning to MISO tuning

In order to solve (3.16), the relationship between λ-parameters and the performance indices of each output, e.g., $\mathrm{OS}_i(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$, and $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ in (3.16), is explored. As performance measures utilized in the controller tuning can not be expressed in an explicit way, an empirical analysis is carried out to assist the tuning procedure.

According to the 2-DOF tuning structure, as the filters F_r and F_d can achieve the same function as the original penalty weights, we have fixed these weighting matrices and adjust the closed-loop performance by $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$. More specifically, as F_d filter appears in the feedback channel, $\boldsymbol{\lambda_d}$ affects the robust stability of the closed-loop system; a larger $\boldsymbol{\lambda_d}$ improves the robustness but also increases the settling time. F_r outside the feedback loop only affects the speed of the reference signal, and therefore the increase of $\boldsymbol{\lambda}$ results in smaller values of overshoot and total variation. As both filters have the diagonal structures, and $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$ only exist on the diagonal entries, it is intuitive to assume that the closed-loop response of ith output is dominated by the ith elements in $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$ (which we refer to as λ_i and $\lambda_{d,i}$ hereafter). Besides, given each output of the MIMO system, the corresponding filters in the F_r and F_d are designed based on the dominant or primary open-loop system producing that output; this also ensures the specific pair of the λ-parameters plays a dominant role only for the closed-loop response of the single corresponding output. In this way, the effect of changing λ_j and $\lambda_{d,j}$ $(j \neq i)$ on $\mathrm{OS}_i(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ and $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ can be ignored. In other words, $\mathrm{OS}_i(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ and $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ can now be represented as $\mathrm{OS}_i(\lambda_i, \lambda_{d,i})$ and $\mathrm{T}_{s,i}(\lambda_i, \lambda_{d,i})$.

These properties are further illustrated in Fig. 3.4 via a typical 2-by-3 MD process; the illustrations are also verified through extensive simulations with other types of MIMO systems, but the results are not shown here due to space limitation. Based on Fig. 3.4, two apparent observations can be obtained: 1) From the right subfigure of Fig. 3.4(a) and that of Fig. 3.4(b), we can see that the adjustment of λ_1 and $\lambda_{d,1}$ does not affect the overshoot and settling time of the second output, which verifies our idea above; 2) the overshoot can be empirically treated as an monotone decreasing function of λ, and the settling time can be considered as an unimodal function of λ_d, which is basically determined by the 2-DOF MPC structure, and is a property that was also found in the SISO MD-MPC tuning [52].

Thus, as the closed-loop response of the ith output is dominated by λ_i and $\lambda_{d,i}$, we can simplify the tuning problem in (3.16) from a MIMO tuning problem to m individual MISO tuning problems, in which we tune λ_i and $\lambda_{d,i}$ for the ith output separately. Mathematically, the new MISO tuning problem



(a) OS of 1st and 2nd output with different λ_1 and $\lambda_{d,1}$

(b) T_s of 1st and 2nd output with different λ_1 and $\lambda_{d,1}$

Figure 3.4: Numerical verification of $\mathrm{OS}_i(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ and $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$

can be stated as

$$\begin{aligned} \min_{\lambda_i, \lambda_{d,i}} \; & \mathrm{T}_{s,i}(\lambda_i, \lambda_{d,i}), \text{for } i = 1, \ldots, m, \\ \text{s.t.} \;\; & \mathrm{OS}_i(\lambda_i, \lambda_{d,i}) \leq \mathrm{OS}_i^* \\ & \mathrm{TV}_i(\lambda_i, \lambda_{d,i}) \leq \mathrm{TV}_i^* \end{aligned} \tag{3.19}$$

based on which the tuning algorithm in the next subsection is proposed. Note that in (3.19) we solve the tuning problem m times for all the MISO distinct form systems.

It is worth noting that different from the overshoot, the total variation is not familiar to the end users, and therefore it is difficult for them to select a proper value for TV_i^* in (3.19). In order to keep the user-friendliness property, which is of high priority here, TV_i^* should be chosen either automatically, or according to other intuitive performance indices (e.g., decay ratio); to achieve this target, the results in last chapter are utilized here to determine the specification of total variation TV_i^* based on the overshoot according to $\mathrm{TV}_i^* = 1 + 2\mathrm{OS}_i^*$, or based on the decay ratio (DR^*) according to $\mathrm{TV}_i^* = 1 + \frac{3\mathrm{OS}_i^*}{1-\mathrm{DR}_i^*}$. Note that overshoot-based TV_i^* is slightly conservative while decay ratio based TV_i^* requires additional knowledge on decay ratio for the end users, and thus the users can choose either one in different situations depending on their preference and process knowledge.

3.4.2 Tuning algorithm

Based on the empirical properties of $\mathrm{OS}_i(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ and $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$, an efficient MIMO tuning algorithm is proposed in this section. As $\mathrm{T}_{s,i}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ is tuned via only $\lambda_i, \lambda_{d,i}$ in the MISO tuning problem (3.19), it is similar to the SISO tuning problem solved in [37]. One feasible method to the tuning problem at hand is to solve each of the MISO tuning problems one by one via the proposed contour-line searching tuning algorithm in [37], but such method is too time consuming to meet the constraint on tuning time, as the overall tuning time equals to the sum of the tuning times for each MISO system; define $t_{\lambda,i}$ as the tuning time for the i^{th} MISO system, the overall tuning time (t_λ) can be represented as

$$t_\lambda = t_{\lambda,1} + t_{\lambda,2} + \cdots + t_{\lambda,m}, \tag{3.20}$$

Based on Algorithm 5, instead of solving the MISO tuning problems one after another, the main idea of the proposed MIMO tuning algorithm here is to solve all the MISO tuning problems simultaneously, by searching for the optimal λ-parameters for all the MISO systems along m identical contour-lines at the same time. Therefore, the proposed tuning algorithm is almost as fast as only solving one MISO tuning problem, and the computation time of the fast algorithm is

$$t_\lambda^f = \max\{t_{\lambda,1}, \ldots, t_{\lambda,m}\} \approx \frac{1}{m} \cdot t_\lambda. \tag{3.21}$$

The fast algorithm is feasible, because the contour-line searching is based on the worst-case performance indices calculated by Algorithm 5 and when the Algorithm 5 is called the performance indices are computed for all the MISO systems. Besides, as all the MISO tuning problems are almost independent (tuning of $(\lambda_i,\ \lambda_{d,i})$ does not significantly affect that of $(\lambda_j,\ \lambda_{d,j}),\ j \neq i$), the proposed tuning algorithm can produce the same results as those of tuning the MISO systems one after another.

In order to develop the multi-counter line MIMO tuning algorithm, we need the following two algorithms: Algorithm 6 identifies the contour lines that $\mathrm{TV}_i(\lambda_i,\ \lambda_{d,i}) = \mathrm{TV}_i^*$ for a fixed set of $\lambda_{d,i},\ i = 1, \ldots, m$ and provides the resultant $\lambda_i,\ i = 1, \ldots, m$ (denoted as λ_i^c) for all the MISO systems; Algorithm 7 searches on the above counter lines for the $\lambda_{d,i},\ i = 1, \ldots, m$ associated with the smallest values of $\mathrm{T_s}$ for all the outputs. The detailed procedures of the MIMO Algorithms 6 and 7 are shown below.

Algorithm 6 Find $\boldsymbol{\lambda^c}(\mathrm{TV}^*, \boldsymbol{\lambda_d})$ and $\mathrm{T}^{\boldsymbol{c}}_{\boldsymbol{s}}(\mathrm{TV}^*, \boldsymbol{\lambda_d})$.

```
Input $\mathrm{TV}^*$, $\boldsymbol{\lambda_d}$, and the uncertainty intervals $[\underline{\boldsymbol{k}}, \overline{\boldsymbol{k}}]$, $[\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}]$ and $[\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}]$ and upper bound for $\boldsymbol{\lambda}$: $\overline{\boldsymbol{\lambda}}^{\boldsymbol{*}}$;
Input $\epsilon$;
**NoFeasibleLambda** $= \mathbf{1}_{1\times m}$;
$\underline{\boldsymbol{\lambda}} \leftarrow \mathbf{0.5}_{1\times m}$, $\overline{\boldsymbol{\lambda}} \leftarrow \overline{\boldsymbol{\lambda}}^{\boldsymbol{*}}$;
**while** $\max(\overline{\boldsymbol{\lambda}} - \underline{\boldsymbol{\lambda}}) > \epsilon$ **do**
    **for** i=1 : m **do**
        $\boldsymbol{\lambda}(i) \leftarrow (\overline{\boldsymbol{\lambda}}(i) + \underline{\boldsymbol{\lambda}}(i)) \times 0.5$
    **end for**
    Numerically evaluate the total variation and overshoot vectors $\mathbf{TV}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ and $\mathbf{OS}(\boldsymbol{\lambda}, \boldsymbol{\lambda_d})$ based on the envelope responses obtained by Algorithm 5.
    **for** i=1 : m **do**
        **if** $\mathbf{TV}(i) > \mathrm{TV}_i^*$ **then**
            $\underline{\boldsymbol{\lambda}}(i) \leftarrow \boldsymbol{\lambda}(i)$;
        **else**
            **if** $\mathbf{OS}(i) < \mathrm{OS}_i^*$ **then**
                $\overline{\boldsymbol{\lambda}}(i) \leftarrow \boldsymbol{\lambda}(i)$;
                **NoFeasibleLambda**$(i) = 0$;
            **else**
                $\underline{\boldsymbol{\lambda}}(i) \leftarrow \boldsymbol{\lambda}(i)$;
            **end if**
        **end if**
    **end for**
**end while**
$\boldsymbol{\lambda^c}(\mathrm{TV}^*, \boldsymbol{\lambda_d}) \leftarrow (\underline{\boldsymbol{\lambda}} + \overline{\boldsymbol{\lambda}})/2$ ;
$\mathrm{T}^{\boldsymbol{c}}_{\boldsymbol{s}}(\mathrm{TV}^*, \boldsymbol{\lambda_d}) \leftarrow \mathrm{T}_{\boldsymbol{s}}(\boldsymbol{\lambda^c}, \boldsymbol{\lambda_d})$;
**end**
```

Algorithm 6: Given a set of $\lambda_{d,i},\ i = 1, \ldots, m$ and the specifications $\mathrm{TV}_i^*,\ i = 1, \ldots, m$, the bisection search method is utilized to find the $\lambda_i^c,\ i = 1, \ldots, m$ on all the counter lines, because each $\mathrm{TV}_i(\lambda_i,\ \lambda_{d,i})$ is a monotone decreasing function of λ_i according to 2-DOF structure; $\mathrm{T}^{\boldsymbol{c}}_{\boldsymbol{s}}(\mathrm{TV}^*, \boldsymbol{\lambda_d})$ here denotes the corresponding worst-case settling times for all the outputs; the

variable **NoFeasibleLambda** indicates the existence of $\boldsymbol{\lambda}$ given a pair of $\boldsymbol{\lambda_d}$ and TV^*. It is worth noting that 1) all the λ_is are tuned before evaluating the TV and OS for all the outputs, based on which the search on all the contour-lines are executed simultaneously, rather than one after another; 2) lines 14-19 are used if the TV^* is chosen based on decay ratio.

Algorithm 7 Tuning of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda_d}$.

```
 1: Input uncertainty intervals $[\underline{\boldsymbol{k}}, \overline{\boldsymbol{k}}]$, $[\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}]$ and $[\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}]$ and upper bound for
    $\boldsymbol{\lambda_d}$: $\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^{\boldsymbol{*}}$;
 2: Input the overshoot specification $\mathrm{OS}^*$;
 3: Calculate the total variation specification according to $\mathrm{TV}^* = 1 + 2\mathrm{OS}^*$
    or $\mathrm{TV}^* = 1 + \frac{3\mathrm{OS}^*}{1-\mathrm{DR}^*}$;
 4: Input $\epsilon$;
 5: $\underline{\boldsymbol{\lambda}}_{\boldsymbol{d}} \leftarrow \mathbf{0.5}_{1\times m}$, $\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}} \leftarrow \overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^{\boldsymbol{*}}$;
 6: while $\max(\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}} - \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}) > \epsilon$ do
 7:     for i=1 : m do
 8:         $\boldsymbol{\lambda_{d1}}(i) \leftarrow \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) + (\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) - \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i)) \times 0.382$;
 9:         $\boldsymbol{\lambda_{d2}}(i) \leftarrow \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) + (\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) - \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i)) \times 0.618$;
10:     end for
11:     Numerically evaluate the settling times $\mathrm{T}^{\boldsymbol{c}}_{\boldsymbol{s1}}(\mathrm{TV}^*, \boldsymbol{\lambda_{d1}})$ and
        $\mathrm{T}^{\boldsymbol{c}}_{\boldsymbol{s2}}(\mathrm{TV}^*, \boldsymbol{\lambda_{d2}})$ based on the Algorithms 5 and 6 ;
12:     for i=1 : m do
13:         if NoFeasibleLambda$(i) = 0$ then
14:             if $\mathrm{T}^c_{s1}(i) > \mathrm{T}^c_{s2}(i)$ then
15:                 $\underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) \leftarrow \boldsymbol{\lambda_{d1}}(i)$;
16:             else
17:                 $\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) \leftarrow \boldsymbol{\lambda_{d2}}(i)$;
18:             end if
19:         else
20:             $\underline{\boldsymbol{\lambda}}_{\boldsymbol{d}}(i) \leftarrow \boldsymbol{\lambda_{d1}}(i)$;
21:         end if
22:     end for
23: end while
24: $\boldsymbol{\lambda_d} \leftarrow (\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}} + \underline{\boldsymbol{\lambda}}_{\boldsymbol{d}})/2$;
25: $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda^c}(\mathrm{TV}^*, \boldsymbol{\lambda_d})$;
26: end
```

Algorithm 7: In the 2-DOF MPC structure, the closed-loop robust stability is determined by the ratio vector $\boldsymbol{\lambda_d}$ according to the sensitivity analysis. Therefore, for each MISO system, the $\mathrm{T}_{s,i}(\lambda^c_i, \lambda_{d,i})$ is approximately a unimodal function of $\lambda_{d,i}$; because an extreme value of the $\lambda_{d,i}$ will result in

either a marginally stable response or a slow response, both of which are associated with a large settling time. Thus, golden search is utilized in Algorithm 7 to obtain the $\lambda_{d,i},\ i = 1, \dots, m$ that offer the smallest worst-case settling times for all the MISO systems; Note that the resultant $\lambda_i,\ i = 1, \dots, m$ and the $\mathrm{T}_{\boldsymbol{s}}^{\boldsymbol{c}}(\mathrm{TV}^*, \boldsymbol{\lambda_d})$ are calculated based on Algorithm 6 as shown in line 11. The robust stability can be achieved based on the proposed algorithms. The reason is that the unstable envelope responses would have large worst-case settling times and thus the corresponding $\lambda_{d,i},\ i = 1, \dots, m$ can not be the optimal solution.

The efficiency of the proposed MIMO tuning technique is shown in Section 3.5.

3.4.3 Prediction of tuning time

The proposed fast tuning algorithm not only solves the MIMO MD-MPC tuning problem efficiently, but also provides a salient feature of accurately predicting the tuning time. Note that an accurate prediction of tuning time is highly demanded for algorithms designed for a commercial tuner software, because the user-friendliness can be improved.

As the proposed tuning algorithm is a line search based method, the computation time can be estimated based on the iteration numbers for convergence and the time required for each iteration. The iteration number only depends on the pre-defined search region and stop criteria, and thus can be calculated based on

$$\frac{\log(\epsilon / \max\left(\overline{\boldsymbol{\lambda}}^* - \mathbf{0.5}_{1\times m}\right))}{\log(0.5)} \tag{3.22}$$

for Algorithm 6 as the bisection search is utilized, and

$$\frac{\log(\epsilon / \max(\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^* - \mathbf{0.5}_{1\times m}))}{\log(0.618)} \tag{3.23}$$

for Algorithm 7 based on the structure of the golden search method. Note that ϵ is the stop criteria utilized in Algorithms 6 and 7 (introduced in line 2 and line 4, respectively), and the constants 0.5 and 0.618 correspond to the convergence rates of the bisection and golden search methods employed in the Algorithms 6 and 7. The iteration time is decided by the time required for numerical evaluating the performance indices, the duration of which can be computed based on the time required to call Algorithm 5, which can also be pre-calculated. Thus, the tuning time can be predicted with Algorithm 8 stated below.

Algorithm 8 Prediction of $t_\lambda(\overline{\boldsymbol{\lambda}}^*, \overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^*)$.

```
1: Input $\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^*$, $\overline{\boldsymbol{\lambda}}^*$ and the uncertainty intervals $[\underline{\boldsymbol{k}}, \overline{\boldsymbol{k}}]$, $[\underline{\boldsymbol{\tau}}, \overline{\boldsymbol{\tau}}]$ and $[\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}]$;
2: Input $\epsilon$;
3: Input $\mathrm{N_{TV}}$ and $\mathrm{N_{T_s}}$;
4: Run the MIMO simulator 3 times to obtain the computation time $t_{\beta,1}$, $t_{\beta,2}$
   and $t_{\beta,3}$;
5: $\bar{t}_\beta = (t_{\beta,1} + t_{\beta,2} + t_{\beta,3})/3$;
6: $x = \frac{\log(\epsilon/\max(\overline{\boldsymbol{\lambda}}^* - \mathbf{0.5}_{1\times m}))}{\log(0.5)}$;
7: $\mathrm{I}_\lambda = \mathrm{ceil}(x)$;
8: $y = \frac{\log(\epsilon/\max(\overline{\boldsymbol{\lambda}}_{\boldsymbol{d}}^* - \mathbf{0.5}_{1\times m}))}{\log(0.618)}$;
9: $\mathrm{I}_{\lambda_d} = \mathrm{ceil}(y)$;
10: $t_\lambda = ((\mathrm{I}_\lambda \times \mathrm{N_{TV}} + \mathrm{N_{T_s}}) \times 2 + \mathrm{I}_{\lambda_d} \times (\mathrm{I}_\lambda \times \mathrm{N_{TV}} + \mathrm{N_{T_s}})) \times \bar{t}_\beta$;
11: end
```

Interpretations for Algorithm 8: $\mathrm{N_{TV}}$ and $\mathrm{N_{T_s}}$ (line 3) are the numbers of the extreme-case systems considered to calculate the total variation and the settling time; according to Algorithm 5 both of them are 8 by default; however, the number of the extreme-case systems considered in Algorithm 5 can be optimized to further save some computation time, and is a future research direction of this work; t_β (line 5) is the time cost for a single run of the simulation (based on which the running time of Algorithm 5 can be obtained), and as the computer normally needs more time to run a program for the first time, the average of three runs is used as the simulation time. I_λ (line 7) and I_{λ_d} (line 9) are the iteration numbers for Algorithm 6 and 7 respectively. The formula in line 10 is constructed based on the structure of the tuning algorithm.

3.5 Industrial Examples

In this section, our algorithm is tested on a model generated based on MD process data obtained from a US paper mill to verify the effectiveness. The 2-by-3 MD process considered is concerned with controlling the Conditioned Weight (CW), and Moisture (M) of the paper products with Base Stock Flow (SF) to the headbox, Main Steam (MS) in the cylinders, and Machine Speed[2]

[2] “Machine speed” indicates how fast the paper machine is running, e.g., 500 feet per minute.

(S) as inputs; specifically, the conditioned weight is controlled by base stock flow and machine speed, while the moisture is controlled by base stock flow and main steam. The nominal model is shown in Table 3.1, and is obtained by the bump test experiment [13] on the original paper making process, in which a number of step changes are first introduced to each input of the paper machine to explore the dynamical property of the system, and based on the experimental data, the nominal model is obtained utilizing a commercial system identification software. Figure 3.5 shows the the step response plot of the nominal process dynamics in the bump test. To illustrate the developed results, the proposed tuning algorithm is applied to the nominal model in this section; also, to evaluate the performance of the resultant MPC controller, the tuning results are tested in the Honeywell real time MPC + Simulator environment.

Table 3.1: Nominal model

O\I	SF	MS	S
CW	$\frac{0.35}{80s+1}e^{-80s}$	-	$\frac{-0.025}{30s+1}e^{-51s}$
M	$\frac{0.4}{90s+1}e^{-80s}$	$\frac{-0.08}{150s+1}e^{-80s}$	-

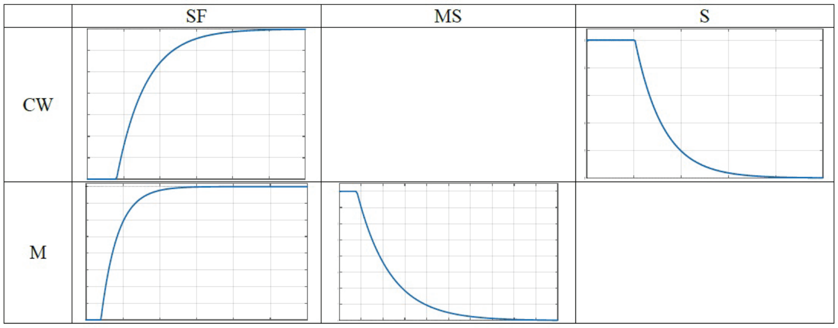


Figure 3.5: Illustration of the 2-by-3 MD process

The MPC penalty weights are $Q_1 = I$, $Q_2 = 0.01 \times I$, $Q_3 = 0$ and the control period is 15 seconds; H_p and H_u are 68 and 27, respectively. Given the multi-variable process in Table 3.1, we have utilized the fast MIMO tuning algorithm to achieve a large number of performance requirements, from OS* = 10% to OS* = 50%, at different types of uncertainty levels, from $[-25\%, 25\%]$ to $[-50\%, 50\%]$. It is worth noting that all the performance specifications and uncertainties are selected based on the industrial experience to cover possible situations that may occur in practice. In order to further evaluate the optimality of the proposed results, we have compared the tuning results of Algorithm 7 with those achieved by the brute force search method[3], given the tuning problem in (3.19). The results of the optimality test for uncertainty level [-50%, 30%] are shown in Table 3.2-3.3. Besides, the actual computation time and the beforehand predicted tuning time by Algorithm 8 are indicated for each case in Table 3.4. The test results for other uncertainty levels are not shown here to save the space. It is worth mentioning that the tuning results from the MIMO algorithm are almost exactly the same as those of brute search, indicating the optimality of the proposed method. The proposed algorithm is also very computational efficient, as for a particular requirement of OS, the tuning time is about 10 seconds[4] while that of the brute force search method is around 20 minutes. In addition, the predicted tuning time from Algorithm 8 produces a very accurate prediction with error less than 5%. The envelope responses of both outputs of this process for OS* = 20% under uncertainty region [-50%, 30%] are also shown in Figure 3.6.

Table 3.2: Comparison for output 1

Overshoot specification	OS*	10%	20%	30%	40%	50%
total variation specification	TV*	1.4	1.8	2.2	2.6	3
Fast tuning algorithm	OS	0.1%	19.03%	27.76%	28.7%	49.08%
	TV	1.26	1.49	2.09	2.50	2.62
	t_s	1200s	1425s	1125s	990s	1110s
	λ_1	2.4413	1.3872	1.1316	1.1872	0.6646
	$\lambda_{d,1}$	0.8970	1.9145	1.2034	0.7793	1.2939
Brute force search method	OS	0.5%	19.03%	27.76%	27.7%	49.08%
	TV	1.32	1.49	2.09	2.41	2.62
	t_s	1175s	1425s	1125s	990s	1110s
	λ_1	2.2131	1.3872	1.1316	1.2157	0.6646
	$\lambda_{d,1}$	0.8810	1.9145	1.2034	0.7841	1.2939

[3] The brute force search method exhaustively tests all possible $\boldsymbol{\lambda}, \boldsymbol{\lambda_d}$ to find the optimal λ-parameters for the tuning problem in (3.19).

[4] The simulation is performed on a computer with Intel i5 CPU and 6G RAM.

Table 3.3: Comparison for output 2

Overshoot specification	OS*	10%	20%	30%	40%	50%
total variation specification	TV*	1.4	1.8	2.2	2.6	3
Fast tuning algorithm	OS	9.59%	17.20%	16.46%	37.56%	41.39%
	TV	1.20	1.64	2.04	2.24	2.44
	t_s	1575s	1335s	1155s	1215s	1185s
	λ_2	1.1444	0.8203	0.8203	0.5646	0.5090
	$\lambda_{d,2}$	1.1657	0.8853	0.6078	0.7793	0.7388
Brute force search method	OS	9.19%	17.20%	16.46%	39.56%	41.39%
	TV	1.20	1.64	2.04	2.41	2.44
	t_s	1585s	1335s	1155s	1200s	1185s
	λ_2	1.1665	0.8203	0.8203	0.5465	0.5090
	$\lambda_{d,2}$	1.1775	0.8853	0.6078	0.7676	0.7388

Table 3.4: Results of predicting tuning time

Overshoot specification	10%	20%	30%	40%	50%
total variation specification	1.4	1.8	2.2	2.6	3
Real tuning time	9.58s	9.56s	9.62s	9.69s	9.69s
Predicted tuning time	9.96s	10.00s	10.1s	9.74s	10.17s

Now the Honeywell real time MPC + Simulator is utilized to further verify the proposed tuning method. The Honeywell real time MPC + simulator environment serves as a virtual realistic testbed for the proposed MPC algorithm - the simulator is utilized to generate the dynamic responses of the MD process subject to disturbances, while the real time MPC is consistent with that implemented in a commercial controller with constraints to generate the control inputs. Different from the nominal model, the real process considered here is shown in Table 3.5, which is chosen based on the uncertainty region

Table 3.5: Process model

O\I	SF	MS	S
CW	$\frac{0.455}{104s+1}e^{-104s}$	-	$\frac{-0.0325}{30s+1}e^{-66.3s}$
M	$\frac{0.52}{117s+1}e^{-104s}$	$\frac{-0.1040}{195s+1}e^{-104s}$	-

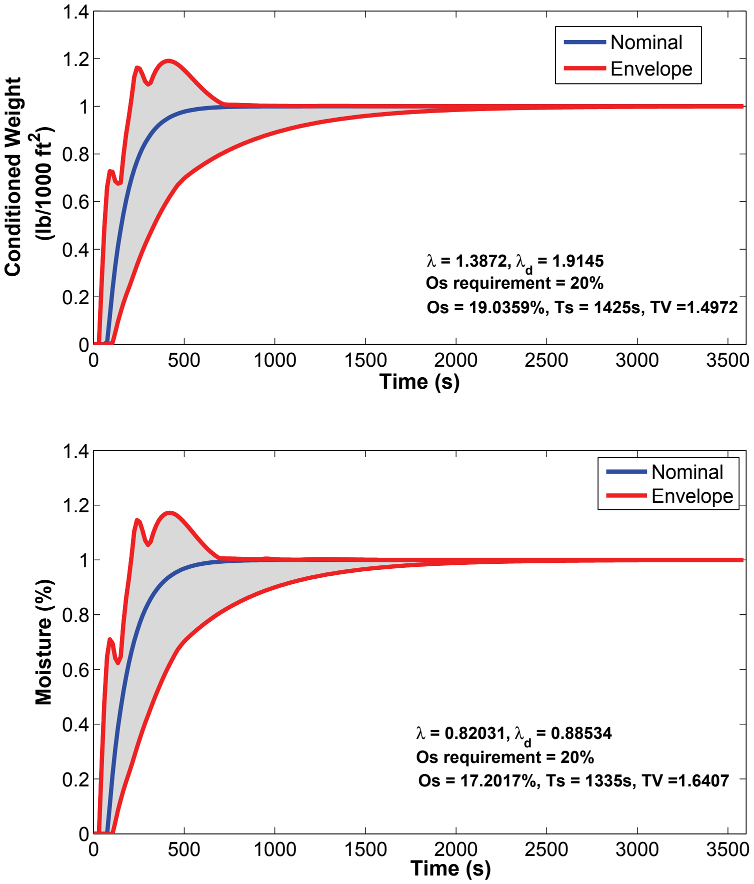


Figure 3.6: Step response envelopes obtained by proposed tuning algorithm

considered in the tuning. $\mathbf{y}(0) = [432\ 8]^{\mathrm{T}}$ and $\mathbf{u}(0) = [3790\ 79\ 447]^{\mathrm{T}}$ are the initial conditions for the controlled and manipulated variables. H_p, H_u, Q_1, Q_2 and Q_3 are selected according to their values used in the aforementioned tuning algorithm, and hard constraints on $\boldsymbol{U}$ and $\boldsymbol{\Delta U}$ utilized in the real time MPC are as follows:

$$\begin{aligned} 0.5\mathbf{u}(0) \leq& \boldsymbol{U} \leq 1.5\mathbf{u}(0), \\ -0.1\mathbf{u}(0) \leq& \boldsymbol{\Delta U} \leq 0.1\mathbf{u}(0), \end{aligned} \tag{3.24}$$

Note that the constraints in (3.24) are considered here for two reasons: 1) to be consistent with the realism of considered process and MPC; 2) to test the performance of the proposed algorithms (in which the constraints are not considered) on the MPC with the real constraints. Besides, we also introduced several changes on the operating conditions based on the situations that may happen in practice; set-point changes with amounts of 1 lbs/1000 ft^2 and

1 % are introduced at $t = 1500s$ for output 1 and 2, simultaneously; a 2 lbs/1000 ft^2 disturbance is made for output 1 at $t = 4800s$, and a -2 % output disturbance is in effect at $t = 2250s$ for output 2; a 10 gpm disturbance is considered at $t = 3300s$ for input 1; and a Gaussian distributed noise with mean 0 and standard deviation 0.1 is introduced to all the measurements. The obtained results for the process outputs and inputs are shown in Figs. 3.7 and 3.8, respectively. In these obtained figures we can see that although a large amount of model uncertainty and different kinds of noises are considered, the closed-loop responses of all the system outputs can still track the reference signals under different working conditions; this further illustrated the promise of the fast multi-variable MPC tuning technique.

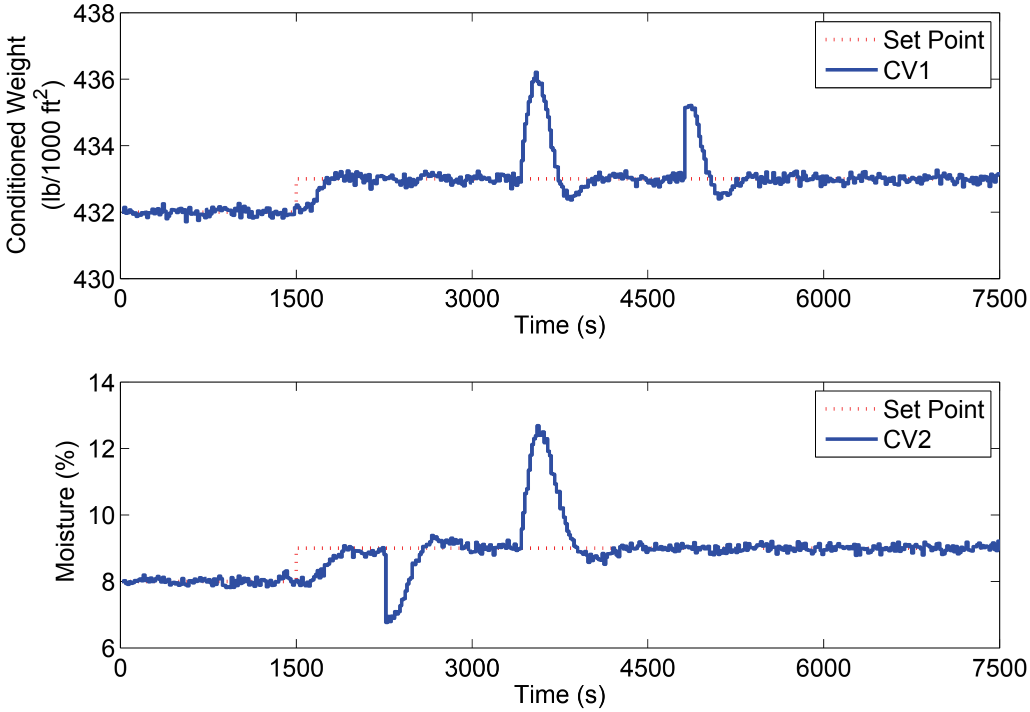


Figure 3.7: Outputs obtained by the Honeywell Real time MPC + Simulator

3.6 Summary

This chapter investigates a robust controller tuning problem for the MD-MPC used in MIMO paper-making processes under user-specified interval uncertainty. An envelope algorithm to evaluate the worst-case OS, T_s and TV

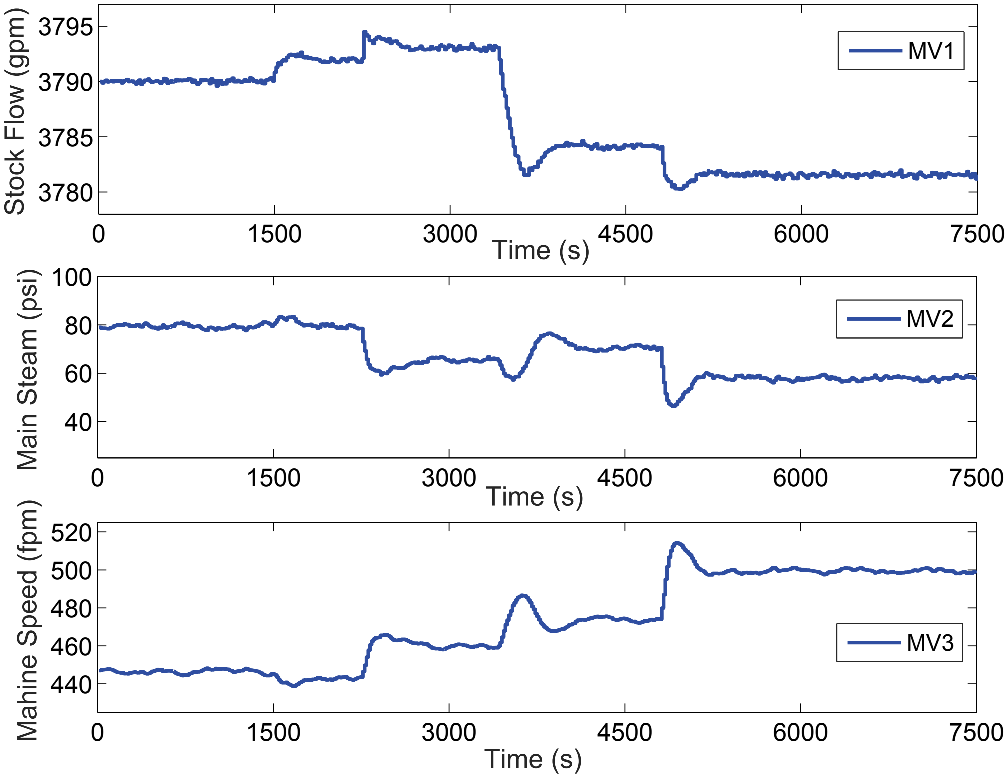


Figure 3.8: Inputs obtained by the Honeywell Real time MPC + Simulator

considering the interval uncertainty for the MIMO MD-MPC is first developed, and then a fast MIMO tuning technique is developed to calculated the MPC tuning parameter vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_d$ that fulfill the requirements on the worst-case time-domain performance indices under the parametric uncertainty for the MIMO system; in addition, a technique to predict the tuning time is also designed to further improve the user-friendliness. The effectiveness of the proposed method is illustrated based on a real MIMO model used by the MD-MPC at a paper mill.

On the other hand, the idea of tuning utilized in this work can be extended to the oscillatory systems (e.g., second-order-plus-time-delay systems), but some modifications and verifications may need to be performed, including 1) the two filters f_r and f_d can still be designed in the first order structure, but an estimated time constant may need to be specified based on the settling time of the oscillatory system's open loop response; 2) the analytical expression of the controller utilized in Algorithm 5 needs to be derived again based on the oscillatory system model; 3) the envelope algorithm utilized to visualize the robust performance needs to be verified on the oscillatory systems with parametric uncertainty; 4) the relationship between the filters and the performance needs to be verified. Note that the modifications in 1)-2) mentioned above are minor and can be achieved easily. Although verification tasks 3)-4) may cost some time, the underlying properties utilized in 3)-4) should hold for the oscillatory systems based on the intuition and the theoretical analysis in this work. Overall, we expect the proposed algorithms to be expendable to the oscillatory systems, and will explore this topic in our future work.

Chapter 4

User-Friendly Robust Spatial Tuning for CD-MPC[*]

4.1 Introduction

In this chapter, we focus on the robust spatial tuning of cross-directional model predictive control under model-plant mismatch. The objective is to tune the MPC weighting matrices such that 1) the robust stability is achieved; 2) the variability of the steady-state actuator and measurement profiles is minimized. As the CD system has a limited spatial bandwidth, the actuation at high spatial frequencies is harmful to the process, and therefore it is essential that the limited spatial bandwidth property be explicitly incorporated into the design of the weighting matrices to suppress the high frequency components in the actuator profile. Besides, the parameters in CD processes are usually identified through bump tests [23, 27, 28] and are inevitably subject to model parameter uncertainties. Thus, it is also desirable that the pre-specified parametric uncertainties be directly considered in the spatial tuning algorithm to guarantee robust performance. To achieve the above mentioned objectives, an automated spatial tuning algorithm is developed in this chapter, and the contributions are as follows:

- A weighting matrix S_b has been appropriately designed via the Fourier matrix approach based on the system model and pre-specified parametric uncertainties. The newly designed S_b penalizes more on the undesirable high-frequency components so that reduced variability of the actuator

[*] The results presented in this chapter were submitted as a provisional U.S. patent [33] and also to IEEE Transactions on Control Systems Technology [32].

and measurement profiles is achieved.

- A systematic automated design procedure has been developed for the spatial tuning of CD-MPC processes. Given a system model and user-specified parametric uncertainties, the matrix S_b and the associated weighting parameters are automatically designed such that robust stability is ensured and better performance is achieved.

4.2 System Description and Preliminaries

In this section, we first introduce each element in the CD-MPC structure (see Fig. 4.1), and then propose some preliminaries on stability and performance analysis.

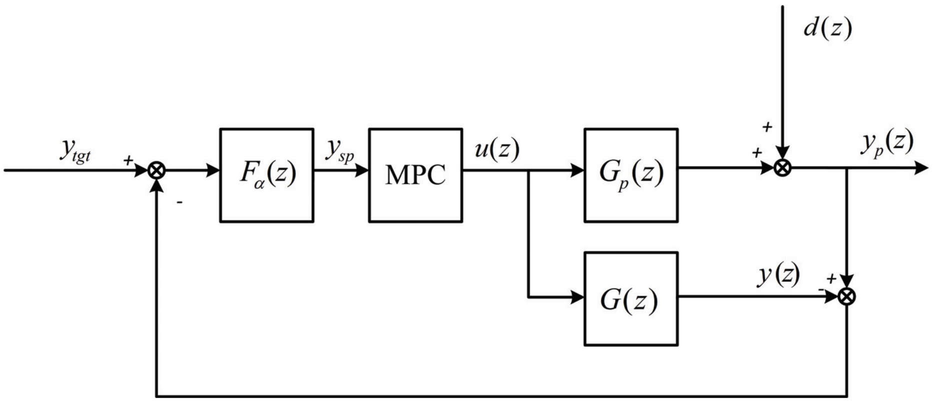


Figure 4.1: The block diagram of the closed-loop CD-MPC system.

As shown in Fig. 4.1, the closed-loop CD-MPC structure for a paper-making process consists of four parts: the real process $G_p(z)$, the nominal model $G(z)$, the MPC and the temporal filter $F_\alpha(z)$, which are illustrated in detail in the following. Moreover, the associated signals are defined as follows: y_{tgt}, y_{sp}, $u(z)$, $d(z)$, $y_p(z)$ and $y(z)$ are the output target, the reference trajectory, the actuator profile, the disturbance profile, the measurement profile, and the predicted output profile, respectively.

4.2.1 Nominal model and model uncertainty

The nominal model $G(z)$ of a CD paper-making process is characterized by

$$G(z) = G_0 h(z), \tag{4.1}$$

where $G_0 \in \mathbb{R}^{m\times n}$ is a constant matrix that characterizes the spatial response/gain of the CD process; $h(z) = (1-a)z^{-t_d}/1 - az^{-1}$ is the temporal transfer function of the process, in which a and t_d are the time constant and time delay in the discretized version.

The spatial gain matrix G_0 has the parameterized structure as shown below [22]:

$$\begin{aligned} G_0 &= [g_1, \ldots, g_n] \in \mathbb{R}^{m\times n}, \\ g_k &= f(x, \gamma, \eta, \xi, \beta, c_k) \\ &= \frac{\gamma}{2}\Bigg\{ e^{-\frac{\eta(x-c_k+\beta\xi)^2}{\xi^2}} \cos\left[\frac{\pi\,(x-c_k+\beta\xi)}{\xi}\right] + \\ &\qquad e^{-\frac{\eta(x-c_k-\beta\xi)^2}{\xi^2}} \cos\left[\frac{\pi\,(x-c_k-\beta\xi)}{\xi}\right] \Bigg\}, \\ &\qquad\qquad x = 1, \ldots, m,\ k = 1, \ldots, n, \end{aligned} \tag{4.2}$$

where γ, η, ξ, and β are the process gain, attenuation, width, and divergence, respectively. They are utilized to characterize the spatial response of each specific actuator. For the k^{th} actuator, c_k is the alignment parameter that determines the center of the corresponding spatial response. The aforementioned model in (4.2) describes the response of each specific actuator in spatial domain and a detailed explanation can be found in [22].

Since model-plant mismatch is unavoidable in process operation and identification, model uncertainties are considered in the work. Based on traditional definitions in robust control to represent model mismatch, it is assumed that the real process model belongs to a set of possible models, characterized by an unstructured or parametric perturbation on the nominal model in (4.1). As parametric uncertainty is easy to understand and specify by the end users in the pulp and paper industry [35, 37], the real process model $G_p(z)$ is described based on the uncertain parameters by

$$
\begin{aligned}
G_p(z) =&\ G_0^p h(z), \\
G_0^p =&\ [g_{1p}, \ldots, g_{np}], \\
g_{kp} =&\ f(x, \gamma_p, \eta_p, \xi_p, \beta_p, c_k^p), \\
\gamma_p =&\ (1 + r_\gamma)\gamma,\ \eta_p = (1 + r_\eta)\eta, \\
\xi_p =&\ (1 + r_\xi)\xi,\ \beta_p = (1 + r_\beta)\beta, \\
c_k^p =&\ c_k + \epsilon, \\
k =&\ 1, \ldots, n,\ x = 1, \ldots, m,
\end{aligned}
\tag{4.3}
$$

where $r_\gamma \in [\underline{r}_\gamma,\ \bar{r}_\gamma]$, $r_\eta \in [\underline{r}_\eta,\ \bar{r}_\eta]$, $r_\xi \in [\underline{r}_\xi,\ \bar{r}_\xi]$, $r_\beta \in [\underline{r}_\beta,\ \bar{r}_\beta]$, and $\epsilon \in [-\underline{\epsilon}\frac{m}{n}, \bar{\epsilon}\frac{m}{n}]$ are utilized to characterize the parametric uncertainties. These trust ranges are also represented as $\gamma_p \in [\underline{\gamma},\ \bar{\gamma}]$, $\eta_p \in [\underline{\eta},\ \bar{\eta}]$, $\xi_p \in [\underline{\xi},\ \bar{\xi}]$, and $\beta_p \in [\underline{\beta},\ \bar{\beta}]$ for brevity; note that as ϵ is a global perturbation on all $c_k^p,\ k = 1, \ldots, n$, $\epsilon \in [\underline{\epsilon},\ \bar{\epsilon}]$ is utilized to represent the trust region of the alignment parameter. Therefore, a set of perturbed models can be characterized by the uncertain model parameters $\gamma_p, \eta_p, \beta_p, \xi_p, \epsilon$. Note that as spatial and temporal components of CD processes are decoupled (the detailed analysis is shown in Section 4.3), only uncertainties on spatial parameters are considered in (4.3) for spatial tuning.

4.2.2 CD-MPC

In this subsection, we start from an industrial CD-MPC controller that has already been successfully applied in paper mills [5]. In this CD-MPC, the following optimization problem is solved:

$$
\begin{aligned}
&\min_{\Delta u(k)} \Bigg\{ \sum_{i=1}^{H_p} (y(k+i) - y_{sp}(k+i))^{\mathrm{T}} Q_1 (y(k+i) \\
&- y_{sp}(k+i)) + \sum_{i=0}^{H_u - 1} \Big[\Delta u(k+i)^{\mathrm{T}} Q_2 \Delta u(k+i) \\
&+ (u(k+i) - u_{sp}(k+i))^{\mathrm{T}} Q_3 (u(k+i) - u_{sp}(k+i)) \\
&+ u(k+i)^{\mathrm{T}} Q_4 u(k+i) \Big] \Bigg\},
\end{aligned}
\tag{4.4}
$$

subject to the system dynamics defined in (4.1) and the constraints as follows:

$$
\Omega \Delta u(k) \leq b - \Gamma u(k-1), \tag{4.5}
$$

where H_p is the prediction horizon, and H_u is the control horizon; $y(k) \in \mathbb{R}^m$ and $y_{sp}(k) \in \mathbb{R}^m$ are the predicted output profile and the corresponding reference signal; $u(k) \in \mathbb{R}^n$ and $u_{sp}(k) \in \mathbb{R}^n$ are the actuator profile and its reference; $\Delta u(k)(= u(k) - u(k-1))$ is the changes in the actuator profile. Q_1 to Q_3 are diagonal weighting matrices; Q_4 is the weighting matrix on the actuator bending/picketing in the following form:

$$
\begin{aligned}
Q_4 &= q_4 S_b^{\mathrm{T}} S_b,\\
S_b &= \begin{bmatrix}
-1 & 1 & 0 & \cdots & \cdots & \cdots & 0\\
1 & -2 & 1 & \ddots & \ddots & \ddots & \vdots\\
0 & 1 & -2 & \ddots & \ddots & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & -2 & 1 & 0\\
\vdots & \ddots & \ddots & \ddots & 1 & -2 & 1\\
0 & \cdots & \cdots & \cdots & 0 & 1 & -1
\end{bmatrix},
\end{aligned}
\tag{4.6}
$$

where q_4 is a scalar weight and $S_b \in \mathbb{R}^{n\times n}$ is the “bending moment matrix” [8, 17]. Note that for the actuator profile, the first and second order derivatives are incorporated in the matrix S_b, and thus the bending behavior is penalized in the cost function of CD-MPC. Ω, Γ and b are the constraint matrices (vectors) derived based on the physical limitations of the process and the details can be found in [5].

Besides, the reference trajectory $y_{sp}(k)$ is generated based on $y_{sp}(k) = F_\alpha(y_{tgt}(k) - d_y(k))$, where $d_y(k) = y_p(k) - y(k)$ and F_α is implemented based on a standard first-order filter [24].

4.2.3 Closed-loop transfer functions

In order to analyze the properties of the CD-MPC shown in (4.4), we employ the method in [20] to calculate the equivalent transfer matrices with the unconstrained MPC, and then evaluate robust stability and performance of the corresponding closed-loop system. To aid the analysis, the closed-loop system can be rearranged in Fig. 4.2. $K_r(z)$ and $K_\alpha(z)$ are derived based on the explicit solution of the unconstrained MPC, following the similar procedure in [20]. Note that as the unconstrained MPC is exploited, the resultant closed-loop system is linear.

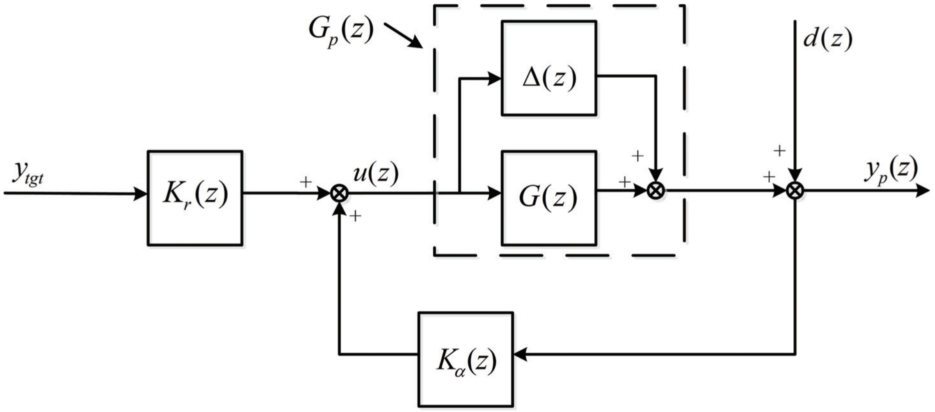


Figure 4.2: The rearranged block diagram of the closed-loop CD-MPC system.

In Fig. 4.2, $G_p(z)$ is the real process that is different from the nominal model $G(z)$, and can be represented in the additive uncertainty form

$$G_p(z) = G(z) + \Delta(z), \tag{4.7}$$

in which $\Delta(z)$ denotes the model uncertainty.

Robust stability is analyzed using the small gain theorem [20]. Specifically, given the parametric uncertainties defined in (4.3), the closed-loop system in Fig. 4.2 is robustly stable for all $G_p(z)$ if it is nominally stable and

$$||T_{ud}(z)\Delta(z)||_\infty < 1 \rightarrow \bar{\sigma}(T_{ud}(e^{i\omega})\Delta(e^{i\omega})) < 1, \forall\omega, \tag{4.8}$$

$$T_{ud}(z) = K_\alpha(z)[I - G(z)K_\alpha(z)]^{-1}, \tag{4.9}$$

where $\bar{\sigma}(A)$ denotes the maximum singular value of a matrix A; $\Delta(z)$ is the model uncertainty in (4.7) and $T_{ud}(z) \in \mathbb{C}^{n\times m}$ is the sensitivity function from the disturbance profile $d(z)$ to the input profile $u(z)$ at the nominal case.

Since the performance of CD control is characterized by its capability to suppress the disturbance, the closed-loop transfer function from the disturbance profile $d(z)$ to the output profile $y(z)$ is used to evaluate the performance

$$T_{yd}(z) = [I - G(z)K_\alpha(z)]^{-1}, \tag{4.10}$$

where $T_{yd}(z) \in \mathbb{C}^{m\times m}$.

$T_{ud}(z)$ and $T_{yd}(z)$ are employed as the key transfer functions in this chapter for evaluating the closed-loop system performance.

Given the CD-MPC structure shown in Fig. 4.1, the closed-loop system behavior is affected by the following parameters: the penalty matrices Q_1 to

Q_4, the temporal filter F_α, the prediction horizon H_p, and the control horizon H_u. Note that in industrial practice, the prediction horizon H_p is normally selected to be 4 times the summation of the time constant and delay; the control horizon $H_u = 1$ is usually utilized in the large-scale MPC (can also be increased based on practical situations); Q_1 is often fixed in the robust tuning. Consequently, only $Q_2 \sim Q_4$, and F_α need to be designed. In next section, we will show that the CD process is a two-dimensional (spatial and temporal) frequency process and the spatial and temporal responses are decoupled, based on which the controller tuning of CD processes can be separated into spatial tuning and temporal tuning.

4.3 Two-Dimensional Frequency Analysis

The spatially invariant property is one of the most important features of CD processes; this allows us to approximate the plant model $G(z)$ as an RCM, and further represent it in the two-dimensional (spatial and temporal) frequency domain. Thus, the multi-variable transfer function $G(z)$ can be simplified into a group of single-input, single-output transfer functions [20]. Denote $\breve{G}(z)$ as the two-dimensional frequency domain representation of $G(z)$, and it can be calculated by

$$\breve{G}(z) = F_m G(z) F_n^{\mathrm{H}}, \tag{4.11}$$

where F_m and F_n are the complex Fourier matrices with dimension m and n, respectively. Then, all the frequency information of $G(z)$ is contained in the following transfer functions:

$$\{\breve{g}(v_1, z), \ldots, \breve{g}(v_n, z)\} = \mathrm{DIAG}(F_m G(z) F_n^{\mathrm{H}}), \tag{4.12}$$

where $\mathrm{DIAG}(A)$ denotes the operation of getting the following elements of a rectangular matrix $A \in \mathbb{C}^{m\times n}$: $\mathrm{DIAG}(A)^1 = \{A(1,1), \ldots, A(k,k), A(k+1+m-n, k+1), \ldots, A(m,n)\}$, where $k = n/2$ if n is even, or $k = (n+1)/2$ if n is odd; and

$$\breve{g}(v_j, z) = \frac{\breve{g}(v_j)(1-a)z^{-t_d}}{1-az^{-1}}, \text{ for } j = 1, \ldots, n. \tag{4.13}$$

Note that based on (4.13), the spatial frequency response gain of $G(z)$ at v_j is $|\breve{g}(v_j)|$, where $v_j,\ j = 1, \ldots, n$, are the spatial frequencies with engineering units and v_n is the actuator Nyquist frequency [20].

[1] $\mathrm{DIAG}(A) = \{A(1,1), \ldots, A(k,k), A(k+1, k+1+m-n), \ldots, A(n,m)\}$ for $A \in \mathbb{C}^{n\times m}$, where $k = n/2$ if n is even, or $k = (n+1)/2$ if n is odd.

Given the special structure of CD processes, it can be further shown that the corresponding sensitivity functions $T_{yd}(z), T_{ud}(z)$ are also RCMs [20]. Thus, they can also be analyzed in the two-dimensional frequency domain as follows:

$$\begin{aligned}
&\breve{T}_{yd}(z) = F_m T_{yd}(z) F_m^{\mathrm{H}}, \\
&\breve{T}_{ud}(z) = F_n T_{ud}(z) F_m^{\mathrm{H}}, \\
&\{\breve{t}_{yd}(v_1, z), \ldots, \breve{t}_{yd}(v_m, z)\} = \mathrm{diag}(\breve{T}_{yd}(z)), \\
&\{\breve{t}_{ud}(v_1, z) \ldots, \breve{t}_{ud}(v_n, z)\} = \mathrm{DIAG}(\breve{T}_{ud}(z)),
\end{aligned} \tag{4.14}$$

where diag(A) represents the diagonal elements of a square matrix A and v_m is the measurement Nyquist frequency. Therefore controller tuning can be implemented based on the sensitivity functions $\breve{t}_{ud}(v, z)$ and $\breve{t}_{yd}(v, z)$ in the two-dimensional frequency domain. Based on the aforementioned analysis and the existing results on CD-MPC tuning for unstructured uncertainty in [18], the tuning task can be separated into two parts: spatial tuning at $z = 1$ ($z = e^{i\omega},\ \omega = 0$) with $Q_3,\ Q_4$, and temporal tuning at $v = 0$ based on Q_2 and F_α.

In the following, the spatial tuning is explored; the corresponding design objective is to tune $Q_3 = q_3 I_n,\ Q_4 = q_4 S_b^{\mathrm{T}} S_b$ so that: (1) the closed-loop system is robustly spatial stable; (2) the variability of the steady-state actuator and measurement profiles is minimized. First a new S_b matrix is designed to improve the spatial performance and then a tuning algorithm is proposed to automatically adjust the weights q_3 and q_4.

4.4 The New S_b to Shape the Spectra of the Actuator Profile

In the existing CD-MPC, the weighting matrix S_b in Q_4 term is a fixed constant matrix disregarding of the CD model considered. In this section, a new S_b matrix, which can tune the frequency response of the actuator profile based on the property of a given CD system, is designed to improve both the nominal and robust performance of the CD-MPC.

4.4.1 The new S_b design for nominal case

In this subsection, we first perform a spatial frequency analysis on the existing S_b and then propose a new S_b to improve the performance. Note that the spatial frequency representation of a signal can be calculated by the Discrete Fourier Transform (DFT), i.e., given a signal $y \in \mathbb{C}^m$, its spatial frequency representation can be obtained by

$$\breve{y} = F_m \cdot y, \tag{4.15}$$

where F_m is the Fourier matrix. To further analyze the signal in the spatial frequency domain, we need the following useful lemma from [20].

Lemma 4.1. *Given spatial signals $y \in \mathbb{C}^m$ and $u \in \mathbb{C}^n$ and assuming $N \in \mathbb{C}^{m\times n}$ is an RCM, if $y = N \cdot u$, then*

$$\breve{y} = \breve{N} \cdot \breve{u}, \tag{4.16}$$

where $\breve{N} = F_m N F_n^{\mathrm{H}}$.

Note that if N is a constant RCM, $\breve{N}$ in Lemma 4.1 denotes its spatial frequency domain representation.

In the cost function of CD-MPC, the cost term associated with Q_4 is

$$u(k+i)^{\mathrm{T}} Q_4 u(k+i) = (S_b u(k+i))^{\mathrm{T}} q_4 (S_b u(k+i)),$$

where the matrix S_b can be approximated as an RCM; its spatial frequency domain representation can be calculated as

$$\{\breve{sb}(v_1), \ldots, \breve{sb}(v_n)\} = \mathrm{diag}(F_n S_b F_n^{\mathrm{H}}). \tag{4.17}$$

Fig. 4.3 illustrates the spatial frequency response (gain) of the existing S_b (namely, $|\breve{sb}(v_j)|$), in which we can see that S_b looks like a high pass filter in the spatial frequency domain. Therefore, when Q_4 penalizes the actuator picketing, based on Lemma 4.1, it actually puts more weights on the high frequency components of the actuator profile, which works like a low pass filter to suppress the high frequency components in the actuator profile u. The reason that an actuator profile with high spatial frequency (namely, actuator picketing) is not desired is that CD processes normally have very small gains at high spatial frequencies (also known as ill-conditioning), and an actuator

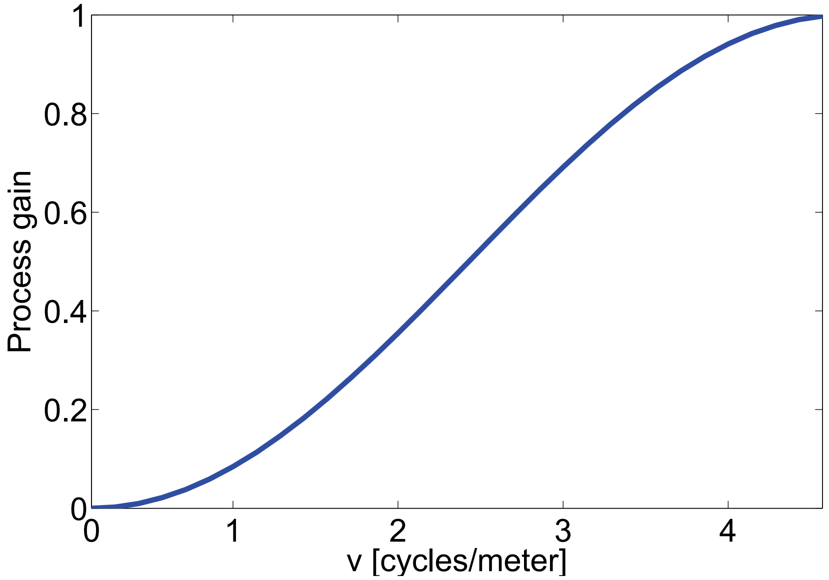


Figure 4.3: The spatial frequency response of the existing S_b.

profile with high spatial frequency could lead to instability of CD processes [5]. However, the spatial frequency response of different CD processes behave differently, and thus, it is beneficial to design a specific S_b for a specific CD process.

To achieve better robustness and performance, the new S_b here is designed to be a high pass spatial filter whose stop band can be adjusted according to different CD processes. To achieve this objective, the definition of cut-off frequency, which is used to characterize the spatial frequency response gain of a CD process, is introduced as follows [15].

Definition 4.1. *Assume the maximum gain of the spatial frequency response of a CD process is g_{max}. The cut-off frequency v_c is the spatial frequency at which the following holds*

$$g_{v_c} = r \cdot g_{max}, r \in (0,\ 0.5]. \tag{4.18}$$

Fig. 4.4 illustrates the above definition using one of the typical single beam CD processes (note that we use $r = 0.1$ in this chapter).

As the process is almost uncontrollable above the cut-off frequency [15], the actuator profile with the spatial frequency components that is higher than v_c is not desirable. Therefore, the stop band of the new S_b can be selected based on v_c of a given CD process, which allows the new S_b to put more

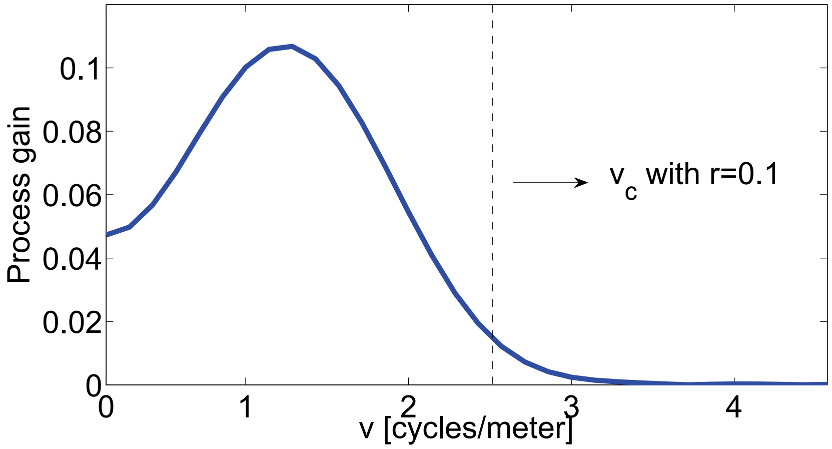


Figure 4.4: The definition of the cut-off frequency v_c.

weights on the high spatial frequency components ($> v_c$) in the input profile while putting small or zero weights on the spatial frequency that is smaller than v_c.

The first approach is to design the new S_b as a high pass Finite Impulsive Response (FIR) filter with the stop band that equals v_c of a given process model. This can be realized by Algorithm 9.

Algorithm 9 Design S_b as a high pass FIR filter with the stop band that equals v_c

```
1: Input $G_0$;
2: Calculate $v_c$ based on $G_0$;
3: Design a low pass filter based on the sinc function and the window function with the stop band that equals $v_c$;
4: Conduct the inverse Fourier transform to obtain the filter in the spatial domain;
5: Construct the desired high pass filter $S_b$ via spectral inversion of the filter obtained in line 4;
6: End
```

Executing Algorithm 9 yields the desired S_b. However, there exist two drawbacks for this method: 1) the stop band of S_b (which we denote as v_b hereafter) may not equal v_c exactly, because the length of the filter is limited by the size of the penalty matrix; 2) even if $v_b = v_c$ can be achieved accurately, the transition band of the filter S_b still needs to be optimized based on the spatial frequency response of $G(z)$.

Considering the aforementioned drawbacks, it is desirable to design S_b with a spectrum that is the mirror of the system's spatial frequency response with respect to the cut-off frequency. Such a S_b matrix can be obtained based on the following idea: we first obtain the mirrored frequency response of a given process model by numerical methods, and then process it to satisfy some conditions and finally perform the inverse Fourier transform using the Fourier matrix, resulting in a S_b with the expected spatial frequency response. Note that the processing step mentioned above is utilized to ensure that the obtained S_b has the circulant property to retain the two-dimensional feature of the closed-loop system and also to guarantee S_b is a real valued matrix for implementation reasons. More specifically, the conditions on the pre-specified spectrum content are summarized in the following to guarantee the desired properties of S_b.

Lemma 4.2. *Define $k_m(j)$, $j = 1, \dots, n$, as the pre-specified spatial frequency response gain of S_b. If $k_m(j) \in \mathbb{R}$, $\forall\ j$ and $k_m(j) = k_m(n-j+2)$ ($j = 2, \dots, (n+1)/2$, if n is odd; or $j = 2, \dots, n/2$, if n is even), then the S_b obtained via the inverse Fourier transform is a real valued circulant matrix.*

Proof. As S_b is a square RCM, the spatial frequency response gains of S_b equal its singular values. Therefore, as the real conjugate even property is guaranteed by $k_m(j) \in \mathbb{R}$, $\forall j$ and $k_m(j) = k_m(n-j+2)$ ($j = 2, \dots, (n+1)/2$, if n is odd; or $j = 2, \dots, n/2$, if n is even), the obtained S_b is a real valued circulant matrix [14]. □

In addition, the desired S_b can also be achieved based on the real valued Fourier matrix [14, 56] to further reduce the computation complexity, and one typical formula for such matrix is shown below.

Definition 4.2. *The real valued Fourier matrix $F_n^r \in \mathbb{R}^{n\times n}$ is defined as*

$$F_n^r(j,k) = \begin{cases} \sqrt{\frac{1}{n}}, & j = 1, \\ \sqrt{\frac{2}{n}} \cdot \sin[(k-1)v_j], & j = 2, \dots, q, \\ \sqrt{\frac{2}{n}} \cdot \cos[(k-1)v_j], & j = q+1, \dots, n, \end{cases} \tag{4.19}$$

where $q = (n+1)/2$ if n is odd, and $q = n/2$ if n is even; $v_j = \frac{2\pi(j-1)}{n}$ is the spatial frequency.

To conclude, the method to generate the improved S_b matrix is summarized in Algorithm 10.

Algorithm 10 Design S_b based on the mirror of the spectrum of the system model

```
1: Input $G_0$;
2: Calculate $v_c$ based on $G_0$;
3: Obtain the desired mirrored spectrum by numerical methods;
4: Process the obtained spectrum content based on Lemma 4.2;
5: Conduct the inverse Fourier transform to obtain the weighting matrix $S_b$
   in the spatial domain;
6: End
```

Fig. 4.5 shows the spatial frequency response of the S_b designed using Algorithm 10 for a typical single beam CD process. The closed-loop performance of the new S_b is illustrated in the next section.

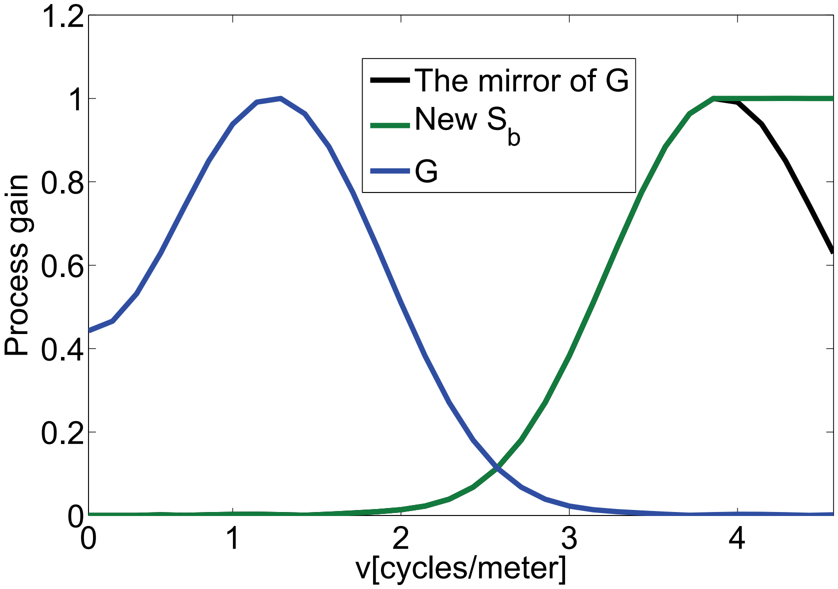


Figure 4.5: The spatial frequency response of the new S_b designed by Algorithm 10.

4.4.2 Performance illustration of the new S_b

As the new S_b is designed based on the frequency property of $G(z)$, it is expected to provide better performance compared with the old S_b. In this section, the performance improvement is illustrated based on the sensitivity analysis.

To analyze the performance of the new S_b, we compare the sensitivity function $T_{yd}(z)$ of the new S_b with that of the old S_b considering the similar robustness (the peak value of T_{ud} is the same). The details are shown in Fig. 4.6.

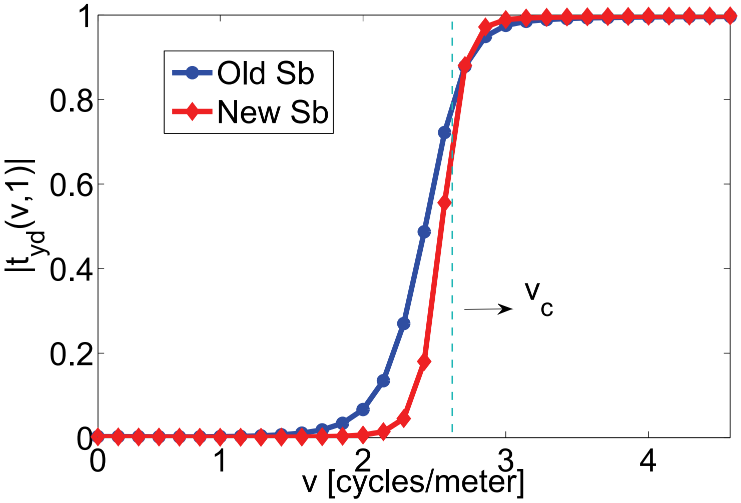


Figure 4.6: Performance comparison of different S_b based on the sensitivity function $T_{yd}(z)$ at steady state ($z = 1$).

Note that the result is only shown at the steady state ($z = 1$) for clear display. We can see that, on the one hand, the sensitivity function with the new S_b has smaller gains at the low frequency, indicating that the disturbance at low frequency (smaller than the cut-off frequency v_c) can be better handled. On the other hand, for the frequency components that are higher than the cut-off v_c, the sensitivity function with the new S_b has larger gains, which prevents the actuator profile from chasing the uncontrollable high frequency components of the disturbance.

It is worth mentioning that if there is no model mismatch, the new S_b can be designed with $v_b = v_c$, and should achieve performance improvement based on the above analysis. However, in the presence of parametric uncertainty, it is better to design v_b to be robust over the model mismatch. To achieve this target, a robust S_b design procedure is introduced in the next section.

4.4.3 The S_b design under parametric uncertainty

In this section, a technique to find the v_b based on the worst spatial model under user-specified parametric uncertainties is proposed, which can help us design a robust S_b that takes the pre-specified model mismatch into account. First, we propose the definition for the worst spatial model as follows:

Definition 4.3. *Given the parametric uncertainty specifications $\eta_p \in [\underline{\eta},\ \bar{\eta}]$, $\beta_p \in [\underline{\beta},\ \bar{\beta}]$, $\gamma_p \in [\underline{\gamma},\ \bar{\gamma}]$, $\xi_p \in [\underline{\xi},\ \bar{\xi}]$, and $\epsilon \in [\underline{\epsilon},\ \bar{\epsilon}]$, the worst spatial model refers to the spatial model that has the smallest cut-off frequency among all the possible models due to parametric uncertainty.*

Based on Definition 4.3, the cut-off frequency of the worst spatial model under parametric uncertainty can be obtained by

$$\begin{aligned} &\min_{\eta_p,\beta_p,\gamma_p,\xi_p,\epsilon} v_c(G_p(z)) \\ &\quad \text{s.t.}\ \ \eta_p \in [\underline{\eta},\ \bar{\eta}],\ \beta_p \in [\underline{\beta},\ \bar{\beta}], \\ &\qquad\qquad \gamma_p \in [\underline{\gamma},\ \bar{\gamma}],\ \xi_p \in [\underline{\xi},\ \bar{\xi}],\ \epsilon \in [\underline{\epsilon},\ \bar{\epsilon}]. \end{aligned} \tag{4.20}$$

Lemma 4.3. *The cut-off frequency v_c for a given process model $G_p(z)$ is independent of the process gain parameter γ_p.*

Proof. As $G_p(z)$ can be approximated as an RCM, v_c can be obtained by $f(x,\eta_p,\beta_p,\gamma_p,\xi_p,c_k^p)$ for any k [15]. Therefore, as γ_p is a multiplication term in $f(x,\eta_p,\beta_p,\gamma_p,\xi_p,c_k^p)$, it does not affect the normalized shape of the spectrum of the system model $G_p(z)$ based on the property of the Fourier transform, and thus is independent of v_c. □

The approach utilized here is to hold all the spatial parameters except one to analyze how the change of this parameter affects the cut-off frequency v_c, and apply similar analysis to the parameters $\eta_p,\beta_p,\gamma_p,\xi_p,\epsilon$ one by one. The results of the aforementioned method are shown in Fig. 4.7 for one CD processes. It is observed that the cut-off frequency v_c is changing monotonically with respect to $\eta_p,\ \beta_p,\ \xi_p,\ \epsilon$ and is not affected by γ_p. This property is also verified through other practical systems, and the details are not shown here due to space limitation.

As $v_c(G_p(z))$ is independent of γ_p, and is changing monotonically with respect to each of $\eta_p,\beta_p,\ \xi_p,\epsilon$, the smallest cut-off frequency can be obtained

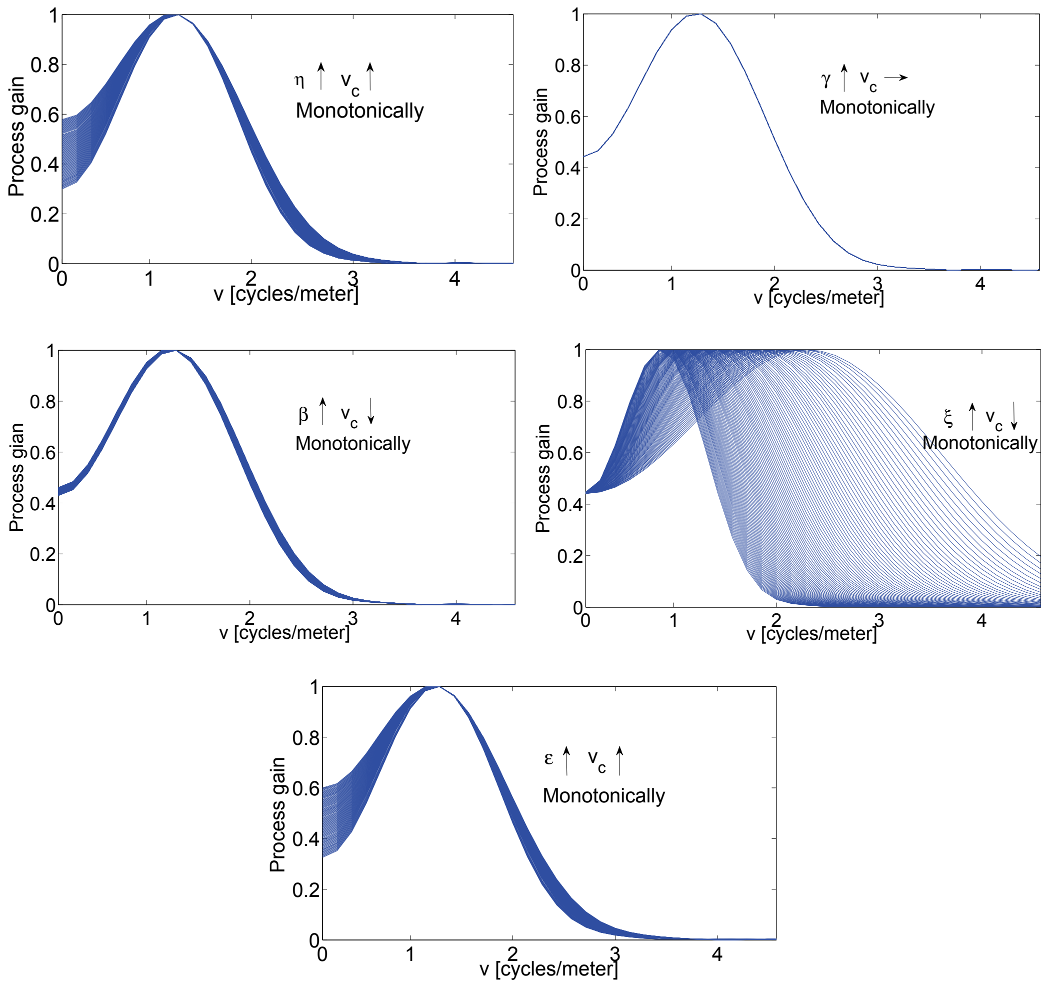


Figure 4.7: The relationship between the spatial parameters and v_c. Note that $\gamma_p, \eta_p, \beta_p, \xi_p$ are increased from 50% to 150% of the nominal values, and ϵ is increased from $-\frac{m}{n}$ to $\frac{m}{n}$.

by solving the following optimization problem:

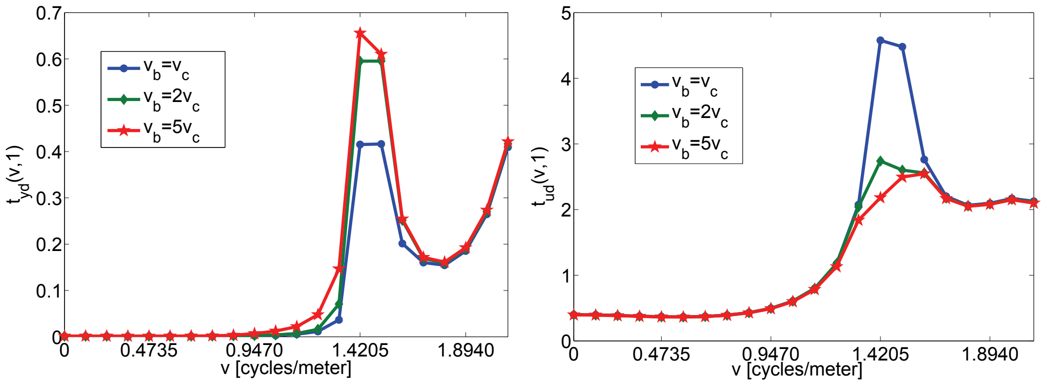
$$\begin{aligned} \min_{\eta_p,\beta_p,\xi_p,\epsilon} \quad & v_c(G_p(z)) \\ \text{s.t.} \quad & \eta_p \in \{\underline{\eta},\ \bar{\eta}\}, \beta_p \in \{\underline{\beta},\ \bar{\beta}\}, \\ & \xi_p \in \{\underline{\xi},\ \bar{\xi}\}, \epsilon \in \{\underline{\epsilon},\ \bar{\epsilon}\}. \end{aligned} \tag{4.21}$$

The obtained worst cut-off frequency is denoted as v_w, and will be utilized in the robust spatial tuning of CD-MPC.

4.5 The Automated Spatial Tuning

The spatial tuning of robust CD-MPC contains two parts: 1) the selection of the stop band v_b of the matrix S_b; 2) the selection of q_3 and q_4. Note that based on the idea in [20], the weight q_3 is selected to be the same as q_4 to simplify the robust tuning problem (this new weight is denoted as q_f).

In order to achieve the tuning objective, the analysis based on the sensitivity functions is utilized to provide some tuning guidelines, and then the automated tuning approach is proposed. Figs. 4.8(a) and 4.8(b) illustrate the effect of changing v_b in the sensitivity functions, and it is observed that a larger v_b gives better robustness, while a smaller v_b gives better performance. If there is no model mismatch, we can select $v_b = v_c$ based on the spatial frequency response of a given process model, the performance of which is also illustrated in Section 4.4. As we can predict the worst spatial model that has the smallest cut-off frequency (v_w) based on the approach in Section 4.4.3, the design of $v_b = v_w$ directly takes care of the given parametric uncertainties, so that a good robustness property can be obtained.

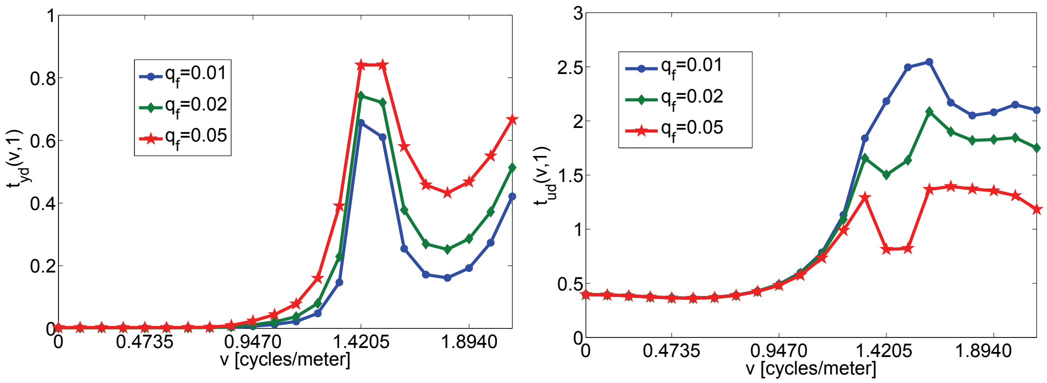


(a)

(b)

Figure 4.8: The sensitivity function $T_{yd}(z)$ (a) and control sensitivity $T_{ud}(z)$ (b) in the spatial frequency domain with $z = 1$: the effect of changing v_b.

Similar analysis can also be applied to the parameter q_f, and the results are shown in Figs. 4.9(a) and 4.9(b). We can observe that a larger q_f gives better robustness, while a smaller q_f gives better performance. As $v_b = v_w$ is selected, the tuning objective of q_f is to achieve robust stability in the spatial domain with $z = 1$. As the robust stability and performance are characterized based on the sensitivity functions, it is useful to derive the sensitivity functions



(a) (b)

Figure 4.9: The sensitivity function $T_{yd}(z)$ (a) and control sensitivity $T_{ud}(z)$ (b) in the spatial frequency domain with $z = 1$: the effect of changing q_f.

in the spatial frequency domain with $z = 1$ (steady state $\omega = 0$), which are obtained as follows:

$$T_{yd}(z = 1) = (I + G_0(Q_3 + Q_4)^{-1}G_m^{\mathrm{T}}Q_1)^{-1}, \tag{4.22}$$

$$T_{ud}(z = 1) = (Q_3 + Q_4 + G_m^{\mathrm{T}}Q_1G_0)^{-1}G_m^{\mathrm{T}}Q_1, \tag{4.23}$$

where $G_m = \sum_{k=1}^{H_p-t_d} \sum_{l=1}^{k} a^{l-1}$.

The tuning objective is then to find the smallest q_f so that the robust stability condition in the spatial frequency domain is satisfied, because it can provide the best performance while still guaranteeing robust stability according to the tuning guidelines in Figs. 4.9(a) and 4.9(b). Based on the robust stability condition in (4.9) and the two-dimensional frequency analysis, the robust stability at $z = 1$ can be achieved if the following condition is satisfied:

$$\bar{\sigma}(T_{ud}(1)\Delta(1)) < 1. \tag{4.24}$$

However, in the presence of parametric uncertainty, the singular values of $\Delta(1)$ are hard to compute directly, and therefore the analysis on the uncertain term is carried out first. From (4.7), the uncertain term $\Delta(1)$ can be represented as

$$\Delta(1) = G_p(1) - G(1). \tag{4.25}$$

Note that as $G_p(1) = G_0^p$ from (4.3), $\Delta(1)$ is only affected by the uncertainties on spatial parameters, i.e., $\eta_p, \beta_p, \gamma_p, \xi_p, \epsilon$. Beside, since uncertainties are specified on the model parameters, $G_p(1)$ is also an RCM based on the method that generates the spatial gain matrix in (4.2). Then, $\Delta(1)$ is also an RCM according to the property that the RCM is closed on the summation operation [20], and therefore it can also be transformed into two-dimensional frequencies

$$\begin{aligned} &\breve{\Delta}(1) = F_m \Delta(1) F_n^{\mathrm{H}}, \\ &\{\delta(v_1, 1), \ldots, \delta(v_n, 1)\} = \mathrm{diag}(\breve{\Delta}(1)). \end{aligned} \tag{4.26}$$

Then, since the singular values equal the gains along the spatial frequencies, equation (4.24) can then be represented in the two-dimensional frequency domain as

$$\max_j(|\breve{t}_{ud}(v_j, 1)\delta(v_j, 1)|) < 1 \rightarrow |\breve{t}_{ud}(v_j, 1)| < \frac{1}{|\delta(v_j, 1)|}, \forall j, \tag{4.27}$$

where $\breve{t}_{ud}(v_j, 1)$ is the sensitivity function $T_{ud}(z)$ with $z = 1$. Existing tuning approaches for large-scale systems normally consider unstructured uncertainty, in which $\Delta(1)$ is a constant for all spatial frequency. In the proposed approach, however, we consider parametric uncertainty to improve user-friendliness, and therefore the key problem here is to calculate the singular values $|\delta(v_j, 1)|, \forall j$, given the user-specified parametric uncertainties in (4.3), which can be mathematically formulated as

$$\begin{aligned} \max_{\eta_p, \beta_p, \gamma_p, \xi_p, \epsilon} \ & |\delta(v_j, 1)|, \text{ for } j = 1, \ldots n \\ \text{s.t.} \ \ & \eta_p \in [\underline{\eta}, \ \bar{\eta}], \beta_p \in [\underline{\beta}, \ \bar{\beta}], \\ & \gamma_p \in [\underline{\gamma}, \ \bar{\gamma}], \xi_p \in [\underline{\xi}, \ \bar{\xi}], \epsilon \in [\underline{\epsilon}, \ \bar{\epsilon}]. \end{aligned} \tag{4.28}$$

It is worth noting that μ analysis can not be applied here because of the large dimensionality of CD processes ($n \leq 200, \ m \leq 2000$) and the nonlinearity of $f(x, \gamma_p, \eta_p, \gamma_p, \xi_p, c_k^p)$.

Due to high nonlinearity of $f(x, \gamma_p, \eta_p, \beta_p, \xi_p, c_k^p)$, based on which $G_p(z)$ and $G(z)$ are generated, explicit representations of the maximum singular values of Δ at $v_j, \ j = 1, \ldots, n$, are hard to achieve. Therefore the following approach is utilized to obtain an approximate solution to the problem in (4.28). Given the temporal frequency $z = 1$, as the singular values of Δ can be obtained by its spatial frequency gains, we can analyze how the spatial gain changes along the spatial frequency with respect to each of the spatial parameters. We increase each of the parameters from its lower bound (50% of the nominal value) to the upper bound (150% times the nominal value) in (4.3) with all the other parameters fixed, and analyze the change of the spatial gains. The results of this analysis are shown in Fig. 4.10 for a typical process, which indicate that the spatial gain at each frequency is decreasing monotonically when a specified parameter is increasing from its lower bound to its nominal value (blue line) while increasing when the parameter increases within the other half of the region (red line).

It is observed that in Fig. 4.10, $|\delta(v_j, 1)|$ for $j = 1, \dots, n$, is changing monotonically with respect to each of $\gamma_p, \eta_p, \beta_p, \xi_p, \epsilon$, so that the extreme values should occur at the boundaries of the uncertainty regions. Thus, the optimization problem in (4.28) is equivalent to:

$$\begin{aligned} \max_{\eta_p,\beta_p,\gamma_p,\xi_p,\epsilon} & \; |\delta(v_j, 1)|, \text{ for } j = 1, \dots n \\ \text{s.t. } & \; \eta_p \in \{\underline{\eta},\ \bar{\eta}\}, \beta_p \in \{\underline{\beta},\ \bar{\beta}\}, \\ & \; \gamma_p \in \{\underline{\gamma},\ \bar{\gamma}\}, \xi_p \in \{\underline{\xi},\ \bar{\xi}\}, \epsilon \in \{\underline{\epsilon},\ \bar{\epsilon}\}. \end{aligned} \tag{4.29}$$

Note that in the optimization problem in (4.29), only the extreme case spatial parameters are considered, which simplifies the problem at hand.

Although the monotonicity property cannot be rigorously proved, it is intuitive in that the maximum singular value, which characterizes the system behavior, usually happens at the largest amount of model mismatch. This is also verified by extensive simulations, in which we compare the maximum singular values of Δ from the proposed approach with that of a large number (> 500) of the Δ's generated randomly within the pre-specified uncertainty region for different processes. Due to space limitation, we only show the results for one of the typical processes, which is a single beam CD process that the dry weight is controlled by the primary autoslice, and the pre-specified parametric uncertainties are

$$\begin{aligned} & \gamma_p \in [0.5\gamma,\ 1.5\gamma],\ \eta_p \in [0.8\eta,\ 1.2\eta], \\ & \beta_p \in [0.8\beta,\ 1.2\beta],\ \xi_p \in [0.5\xi,\ 1.5\xi], \\ & \epsilon \in [-0.6\frac{m}{n},\ 0.6\frac{m}{n}], \end{aligned} \tag{4.30}$$

where $\gamma = 0.26434, \eta = 2.3, \xi = 0.3, \beta = 367, m = 264, n = 44$. The results are shown in Fig. 4.11, in which the blue curves are the singular values of randomly generated systems along the spatial frequency while the red bound is the maximum singular values calculated by solving the optimization problem

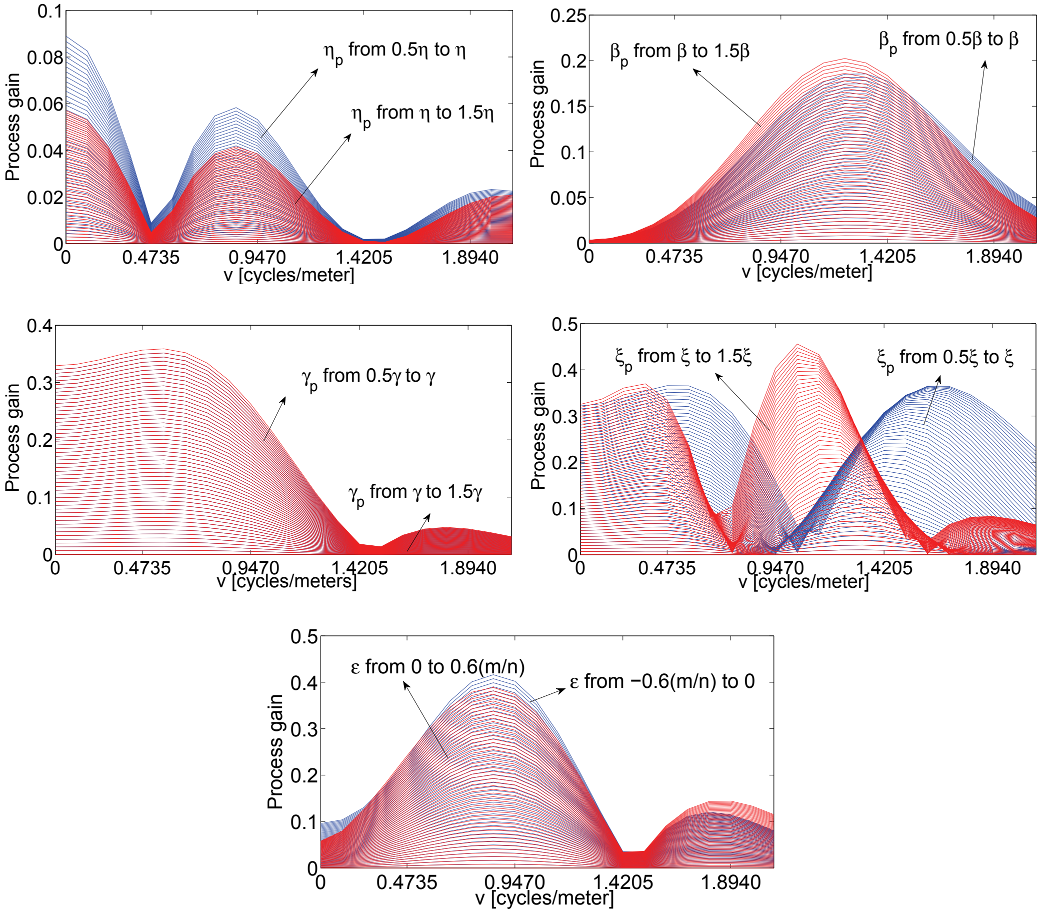


Figure 4.10: The relationship between the spatial parameters and the spatial gain $|\delta(v_j, 1)|$.

in (4.29). We can see that the red bound is a tight bound for all possible singular values due to model mismatch.

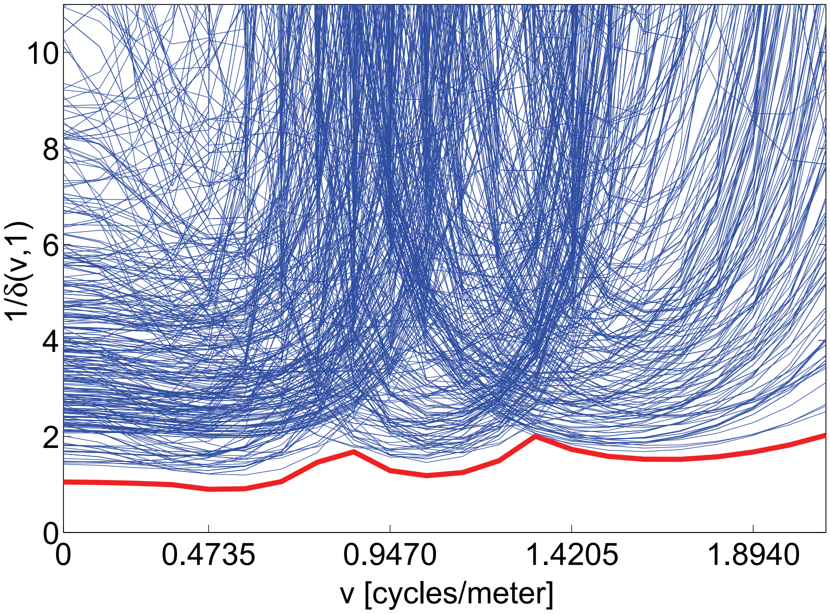


Figure 4.11: Performance illustration of the maximum singular values calculated based on the optimization problem in (4.29).

Based on the tuning guideline of q_f and the robust stability condition in (4.27), the desired q_f can be found by utilizing a bisection search for the smallest q_f that satisfies (4.27), which can be normally completed in about 2 seconds on a desktop with Intel i5 core and 6G memory.

In summary, the automated robust spatial tuning can be implemented according to the flow chart in Fig. 4.12.

4.6 Simulation Results with Real-time Simulator

In this section, we apply the proposed new S_b and the automated robust tuning algorithms to a process model obtained from a paper mill in Canada. This is a single beam process in which the dry weight of the output profile is controlled by the primary autoslice.

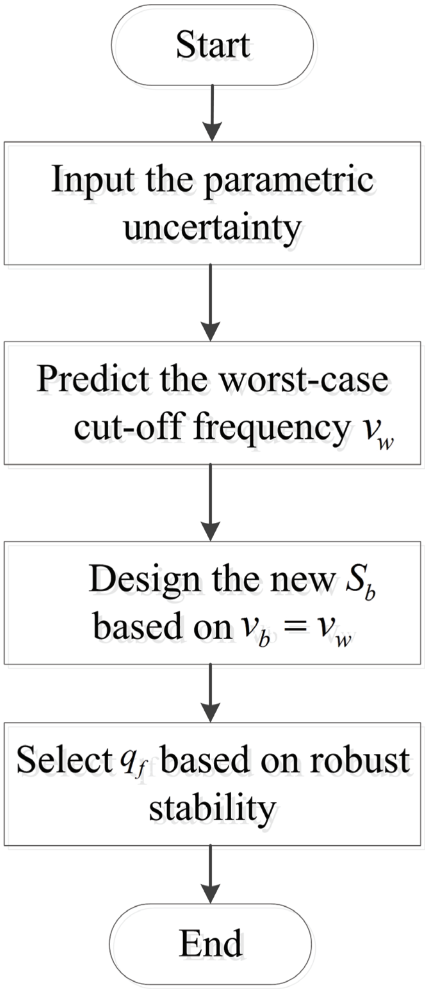

Start
Input the parametric uncertainty
Predict the worst-case cut-off frequency v_w
Design the new S_b based on $v_b = v_w$
Select q_f based on robust stability
End

Figure 4.12: Flow chart of the automated spatial tuning.

In this test, the parametric uncertainty specifications are given as

$$\begin{aligned} &\gamma_p \in [0.75\gamma,\ 1.25\gamma],\ \eta_p \in [0.75\eta,\ 1.25\eta], \\ &\beta_p \in [0.75\beta,\ 1.25\beta],\ \xi_p \in [0.75\xi,\ 1.25\xi], \\ &\epsilon \in [-0.6\frac{m}{n},\ 0.6\frac{m}{n}], \end{aligned} \tag{4.31}$$

where $\gamma = 0.26434, \eta = 2.3, \xi = 0.3, \beta = 367, m = 264, n = 44$.

By utilizing the technique in (4.21), the worst cut-off frequency v_w of the process model for the given parametric uncertainties in (4.31) can be obtained, and then the spectra penalization weighting matrix S_b is designed based on the stop band $v_b = v_w$.

The overall tuning costs 2 seconds on a computer with Intel i5 core and 6G memory. The spatial tuning results in a q_f that equals 0.026 (see Fig. 4.13, which shows that the stability condition in (4.27) is satisfied).

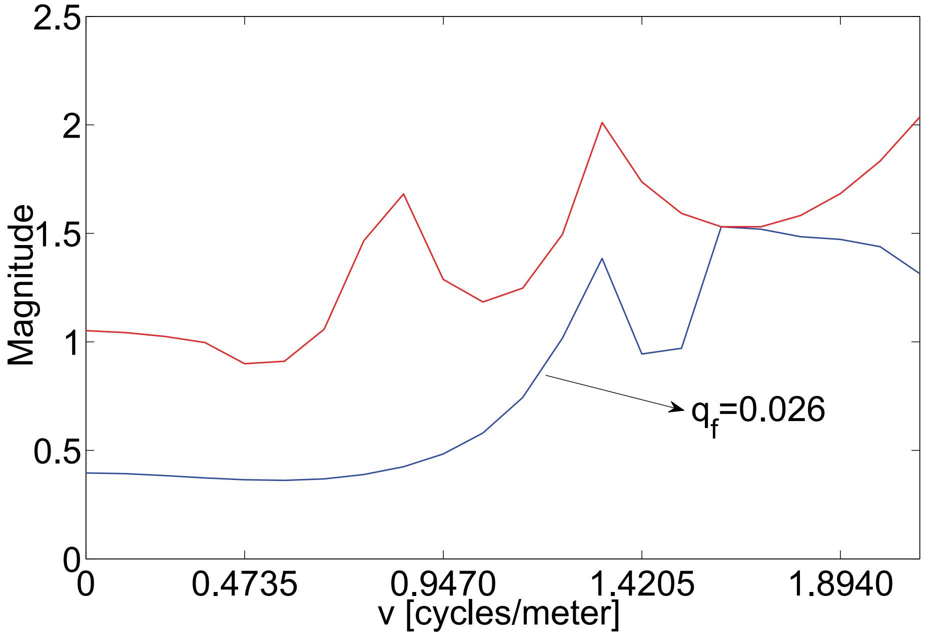


Figure 4.13: Spatial tuning result for the proposed approach. Note that the red line is for the $|\frac{1}{\delta(v,1)}|$ and the blue line is for the $|\breve{t}_{ud}(v,1)|$.

We then apply the obtained tuning results to the Honeywell real time MPC + simulator environment and compare the obtained results with that of the existing controller. Note that the real process model G_p is selected satisfying (4.31). The performance is shown in Figs. 4.6. In Fig. 4.6, the upper figure shows the steady-state measurement profile, while the lower figure is for the corresponding actuator profile. In both of the figures we can see that the profiles using the proposed method (red line and bars) are much better than that of the existing method (gray line and bars) as the input/output process is less oscillatory. More specifically, the 2σ (two times of the standard deviation) of the steady-state actuator profile has been improved by about 66% while that of the measurement profile has been improved by about 44%.

4.7 Summary

An automated spatial tuning approach for cross-directional model predictive control under user-specified parametric uncertainties is proposed in this chapter. A computational efficient approach is first proposed to calculate the

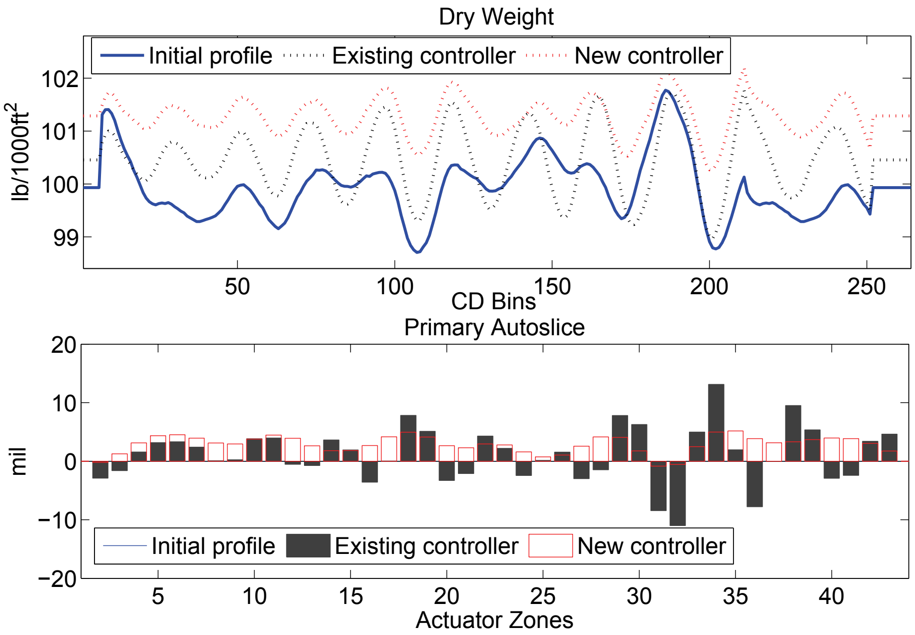


Figure 4.14: Performance illustration for the proposed robust tuning method.

smallest cut-off frequency of all process models given the pre-specified parametric uncertainty. The weighing matrix S_b is then appropriate designed to penalize undesirable high-frequency components in the actuator profile based on the worst-case cutoff to reduce the variability of the actuator and measurement profiles in the spatial domain; finally, an automated tuning method is established to adjust the multiplier of the spatial frequency weighted actuator variability term in the MPC cost function to assure robust spatial stability. The proposed algorithms are validated using a system model extracted from the pulp and paper industry.

Chapter 5

User-Friendly Robust Temporal Tuning for CD-MPC[*]

5.1 Introduction

In this chapter, the temporal tuning problem of cross-directional model predictive control under model-plant mismatch is studied. The aim is to automatically determine the MPC tuning parameters such that 1) the robust stability can be guaranteed; 2) the required time domain performance indices, such as overshoots and settling times, can be satisfied. As CD processes have hundreds of inputs and outputs, it is massive to visualize the indices for all the inputs/outputs simultaneously, and therefore it is more user-friendly to find some novel indices that can both characterize the traditional time domain performance and are also easy to visualize. Besides, as uncertainties on model parameters are unavoidable in system identification, it is desired that the user-specified parametric uncertainties be explicitly accounted in the tuning procedure. In order to achieve the aforementioned objectives, a temporal filter is adopted to smooth the MPC reference trajectory, and a systematic procedure is developed to tune the parameter in the temporal filter. In summary, the contributions of the chapter are as follows:

- the user-friendly temporal performance indices are defined based on the two times of the standard deviation (2σ) of the input/output profile and a visualization technique is proposed to evaluate all the possible 2σ performance given the pre-specified parametric uncertainty.

[*] The results presented in this chapter were submitted as a provisional U.S. patents [34] and also to IEEE Transactions on Control Systems Technology [32].

- an automatic tuning algorithm for the parameter in the temporal filter is presented to guarantee satisfactory performance in terms of the proposed 2σ indices. In the presence of parametric uncertainties, the tuning is based on the worst-case situation, which ensures that the required robust performance can be achieved. Besides, the effectiveness of the proposed tuning algorithm is verified using a system model extracted from the pulp and paper industry.

5.2 Preliminary and Tuning Objective

In this section, we first introduce each element in the CD-MPC structure (see Fig. 5.1) for temporal tuning, and then state the problem to be solved.

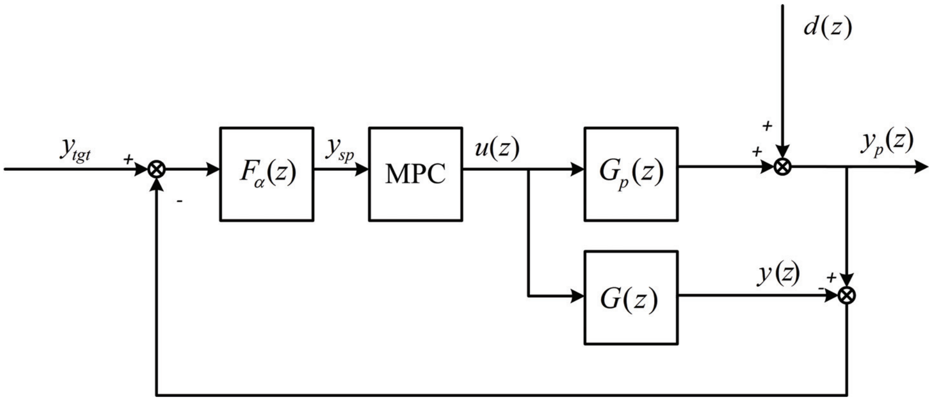


Figure 5.1: The block diagram of CD-MPC structure.

5.2.1 Nominal model and model uncertainty

The nominal model $G(z)$ of a CD paper-making process is characterized by

$$G(z) = G_0 h(z),\ h(z) = \frac{(1-a)z^{-t_d}}{1-az^{-1}}, \tag{5.1}$$

where $G_0 \in \mathbb{R}^{m\times n}$ is a constant matrix that characterizes the spatial response/gain of the CD process [5]. $h(z)$ is the temporal transfer function of the process; a and t_d are the discretized time constant and time delay, which are identified from the input and output data of the real process through bump test experiment [15, 33].

Model-plant mismatch is unavoidable during process identification procedure, and therefore model uncertainties must be considered in the tuning algorithm design. Two different kinds of uncertainties are normally used in robust control theory, namely, parametric uncertainty and unstructured uncertainty. Since unstructured uncertainty is not familiar to practitioners in the pulp and paper industry [37, 53], parametric uncertainty is employed and the real process model $G_p(z)$ is denoted in the following form

$$\begin{aligned} G_p(z) =& \ G_0 h_p(z), \\ h_p(z) =& \ \frac{(1-a_p)z^{-t_{dp}}}{1-a_p z^{-1}}, \\ a_p =& \ (1+r_a)a, \ t_{dp} = (1+r_{t_d})t_d, \end{aligned} \tag{5.2}$$

where $r_a \in [\underline{r}_a, \ \bar{r}_a]$ and $r_{t_d} \in [\underline{r}_{t_d}, \ \bar{r}_{t_d}]$ are utilized to characterize the parametric uncertainties. These trust ranges are also represented as $a_p \in [\underline{a}, \ \bar{a}]$, and $t_{dp} \in [\underline{t}_d, \ \bar{t}_d]$ for brevity; Therefore, a set of perturbed models can be characterized by the uncertain model parameters a_p, t_{dp}. Note that as the spatial and temporal tuning are independent based on the analysis in last chapter, only uncertainties on temporal parameters are considered for temporal tuning.

5.2.2 CD model predictive controller

The MPC controller shown in Fig. 5.1 basically solves the following finite-horizon optimal control problem:

$$\begin{aligned} \min_{\Delta u(k)} \Bigg\{ & \sum_{i=1}^{H_p} (y(k+i) - y_{sp}(k+i))^{\mathrm{T}} Q_1 (y(k+i) \\ & - y_{sp}(k+i)) + \sum_{i=0}^{H_u-1} \Big[\Delta u(k+i)^{\mathrm{T}} Q_2 \Delta u(k+i) \\ & + (u(k+i) - u_{sp}(k+i))^{\mathrm{T}} Q_3 (u(k+i) - u_{sp}(k+i)) \\ & + u(k+i)^{\mathrm{T}} Q_4 u(k+i) \Big] \Bigg\}, \end{aligned} \tag{5.3}$$

subject to the system dynamics defined in (5.1) and the constraints as follows:

$$\Omega \Delta u(k) \leq b - \Gamma u(k-1), \tag{5.4}$$

where

H_p: prediction horizon;

H_u**:** control horizon;

$y(k)$**:** predicted output profile at time k;

$y_{sp}(k)$**:** reference signal for output profile at time k;

$u(k)$**:** actuator profile at time k;

$u_{sp}(k)$**:** reference signal for actuator profile at time k;

$\Delta u(k)$**:** changes in the actuator profile at time k.

$Q_1 \sim Q_4$**:** weighting matrices;

Ω, Γ, b : matrices (vectors) for actuator constraints.

5.2.3 Temporal filter

The traditional MPC output reference trajectory is constructed as a step change, which requires the predicted output profile to track the output target immediately after the dead time of the process. However, in this work, the temporal filter designed in [24] is utilized to generate the reference trajectory $Y_{sp}(k)$ based on

$$Y_{sp}(k) = \begin{bmatrix} y_{sp}(k+1) \\ y_{sp}(k+2) \\ \vdots \\ y_{sp}(k+H_p) \end{bmatrix} = F_\alpha(y_{tgt}(k) - d_y(k)), \tag{5.5}$$

where $y_{tgt}(k)$ is the output target, and $d_y(k) = y_p(k) - y(k)$ is the disturbance estimated based on the process output $y_p(k)$ and predicted output $y(k)$. F_α is the time domain implementation of $f_\alpha(z)$ based on $y_{sp}(z) = f_\alpha(z) I_m (y_{tgt}(z) - d_y(z))$ and $f_\alpha(z)$ is the temporal filter

$$f_\alpha(z) = \frac{(1-a_r)z^{-t_d}}{1 - a_r z^{-1}}, \tag{5.6}$$

where $a_r = e^{-\frac{\Delta T}{\alpha \tau}}$; ΔT is the sampling time and τ is the continuous-time time constant of the temporal transfer function of the process; I_m represents an m-by-m identity matrix. Based on this filter, the aggressiveness of the control signal can be adjusted by the parameter α with Q_2 set to a small-valued scalar matrix. It is worth noting that the function of Q_2 can be achieved by parameter α. The reason is that parameter α can be adjusted to filter the reference signal of the MPC so that there is no aggressive change in the reference trajectory of the output profile, and correspondingly the aggressive control signal can be avoided. On the other hand, α-based tuning is more intuitive to the end users, because, different from the original tuning parameter Q_2 which seems not to have a direct meaning for its value, the value of parameter α can provide an approximation on the closed-loop time constant based on the open-loop one [24].

5.2.4 Tuning objective

Given the temporal parametric uncertainties defined in (5.2), the real process $G_p(z)$ can also be represented in the following form

$$G_p(z) = G(z) + \Delta(z), \tag{5.7}$$

where $\Delta(z)$ indicates the additive uncertainty. Compared with the robust spatial tuning, the only difference in the closed-loop system here is the model uncertainty term $\Delta(z)$ in $G_p(z)$[1], and therefore the closed-loop transfer functions for temporal tuning are the same as those of the spatial tuning. Therefore, given the parametric uncertainties in (5.2), the closed-loop system in Fig. 5.1 is robustly stable for all $G_p(z)$ if it is nominally stable and

$$||T_{ud}(z)\Delta(z)||_\infty < 1 \rightarrow \bar{\sigma}(T_{ud}(e^{i\omega})\Delta(e^{i\omega})) < 1, \forall\omega, \tag{5.8}$$

where $\bar{\sigma}(A)$ indicates the maximum singular value of a matrix A; $T_{ud}(z) \in \mathbb{C}^{n\times m}$ is the sensitivity function from the disturbance profile $d(z)$ to the input profile $u(z)$ at the nominal case. Besides, the closed-loop transfer function from the disturbance profile $d(z)$ to the output profile $y(z)$, namely, $T_{yd}(z) \in \mathbb{C}^{m\times m}$, is also utilized to evaluate the performance. Based on the two-dimensional analysis in last chapter, the controller tuning can be implemented based on the sensitivity functions $\breve{t}_{ud}(v, z)$ and $\breve{t}_{yd}(v, z)$ in the two-dimensional frequency domain, and furthermore, temporal tuning can be considered only at $v = 0$ via the parameter α in the temporal filter.

Based on the aforementioned analysis, the temporal tuning of CD-MPC under pre-specified parametric uncertainty is considered in the following, and

[1] The reason is the uncertainties are specified on different parameters.

the corresponding design objective is to tune the parameter α so that: (1) the closed-loop system is robustly stable at $v = 0$; (2) the time domain performance indices, namely, settling times and overshoots, meet the end users' specifications.

5.3 Robust Stability Analysis and Performance Visualization

In this section, we first calculate a lower bound of the tuning parameter α to guarantee robust stability at the spatial frequency $v = 0$, and then develop a visualization technique for robust performance.

5.3.1 Robust stability analysis

Similar to the spatial tuning, the robust stability condition for temporal tuning can be represented as

$$|\breve{t}_{ud}(0, e^{i\omega})\delta(0, e^{i\omega})| < 1, \forall\omega \ \rightarrow |\breve{t}_{ud}(0, e^{i\omega})| < \frac{1}{|\delta(0, e^{i\omega})|}, \forall\omega. \tag{5.9}$$

where $\breve{t}_{ud}(0, e^{i\omega})$ and $\delta(0, e^{i\omega})$ are $T_{ud}(z)$ and $\Delta(z)$ at spatial frequency $v = 0$.

In order to achieve (5.9), it is essential to find the maximum value of $|\delta(0, e^{i\omega})|$. Based on the parametric uncertainties in (5.2), this can be achieved by solving

$$\begin{aligned} &\max_{a_p, t_{dp}} |\delta(0, e^{i\omega})|, \ \forall\omega \\ &\text{s.t.} \ \ a_p \in [\underline{a}, \ \bar{a}], t_{dp} \in [\underline{t}_d, \ \bar{t}_d]. \end{aligned} \tag{5.10}$$

Given the constrained optimization problem in (5.10), the optimal solution can be obtained by examining every possible combination of active constraints according to the Karush-Kuhn-Tucker condition, which, however, requires too much computationally time for the problem at hand. Based on the analysis and numerical verification in [36, 37], the extreme system behavior normally happens at the extreme model parameters, and therefore the maximum value of $|\delta(0, e^{i\omega})|$ can be obtained by solving

$$\begin{aligned} &\max_{a_p, t_{dp}} |\delta(0, e^{i\omega})|, \ \forall\omega \\ &\text{s.t.} \ \ a_p \in \{\underline{a}, \ \bar{a}\}, t_{dp} \in \{\underline{t}_d, \ \bar{t}_d\}. \end{aligned} \tag{5.11}$$

Then, a bisection search can be utilized for the smallest α that satisfies (5.9), as increasing α can provide better robustness (i.e., $\alpha \uparrow$ leads to $|\breve{t}_{ud}(0, e^{i\omega})| \downarrow$). Then, the obtained α is a lower bound for α that guarantees robust stability, and is denoted as α^* hereafter. Note that only a finite number of temporal frequencies distributed from 0 to the Nyquist frequency may be considered in solving (5.11) due to the implementation issue [18].

5.3.2 User-friendly performance indices and their visualization

In this subsection, we focus on robust performance. As existing robust tuning approaches normally consider frequency domain performance evaluation, which may not be intuitive to end users in the pulp and paper industry [53], the time domain performance indices, e.g., settling times and overshoots, are utilized. The traditional time domain indices are characterized based on the step response of each output measurement, which, however, is not applicable here due to the fact that CD processes, even for single beam cases, have hundreds of output measurements (which we call the measurement profile). Therefore, it is more intuitive and efficient to transform m dimensional output measurements into just one dimension and then to adopt the time domain performance indices. In this work, we select the spread of the two times of the standard deviation (2σ) of the input/output profile as the one dimensional representation which is widely used in the pulp and paper industry and is familiar to end users. Before proceeding, the temporal tuning indices are defined as follows.

Definition 5.1 (Overshoot of the 2σ spread)**.** *The overshoot of a stable 2σ spread is its maximum value minus the final value divided by the final value.*

Definition 5.2 (Settling time of the 2σ spread)**.** *The settling time of a stable 2σ spread is the time required for the spread to reach and stay at its final value.*

As uncertainties are considered on the dynamic parameters a and t_d, which results in a set of process models, an efficient visualization technique needs to be designed to evaluate all the possible 2σ spreads to calculate the proposed performance indices. This visualization for the output 2σ spread can be obtained by solving the following optimization problems:

$$\begin{aligned} &\min_{a_p, t_{dp}} 2\sigma(y_i(a_p, t_{dp})), \ \text{for } i = 1, \ldots, l \\ &\text{s.t. } \ a_p \in [\underline{a}, \ \bar{a}], \ t_{dp} \in [\underline{t}_d, \ \bar{t}_d], \end{aligned} \tag{5.12}$$

$$\begin{aligned} &\max_{a_p, t_{dp}} 2\sigma(y_i(a_p, t_{dp})), \ \text{for } i = 1, \ldots, l \\ &\text{s.t. } \ a_p \in [\underline{a}, \ \bar{a}], \ t_{dp} \in [\underline{t}_d, \ \bar{t}_d], \end{aligned} \tag{5.13}$$

where l is the entire scan time for the process. Note that as the spatial gain matrix G_0 only affects the steady-state value of the 2σ, it is selected to be the worst spatial model obtained in Chapter 4, which produces the worst spatial performance (on the basis of the steady-state values of the 2σs).

Based on the FOPDT structure of $h(z)$, the extreme closed-loop system behavior normally happens when all the extreme parameters are taken simultaneously [36, 53], and therefore the above optimization problems can be approximately solved only at $a_p \in \{\underline{a}, \ \bar{a}\}$ and $t_{dp} \in \{\underline{t}_d, \ \bar{t}_d\}$ to obtain the envelope of the 2σ spreads. More specifically, the robust performance visualization can be achieved with

$$\begin{aligned} &\min_{a_p, t_{dp}} 2\sigma(y_i(a_p, t_{dp})), \ \text{for } i = 1, \ldots, l \\ &\text{s.t. } \ a_p \in \{\underline{a}, \ \bar{a}\}, \ t_{dp} \in \{\underline{t}_d, \ \bar{t}_d\}, \end{aligned} \tag{5.14}$$

$$\begin{aligned} &\max_{a_p, t_{dp}} 2\sigma(y_i(a_p, t_{dp})), \ \text{for } i = 1, \ldots, l \\ &\text{s.t. } \ a_p \in \{\underline{a}, \ \bar{a}\}, \ t_{dp} \in \{\underline{t}_d, \ \bar{t}_d\}. \end{aligned} \tag{5.15}$$

Similarly, this approach can also be applied on the input profile to achieve visualization. The performance of the visualization method based on optimization problems in (5.14) and (5.15) is verified through many industrial examples. To save space, we only show the result (see Fig. 5.2) for one typical CD process (dry weight controlled by primary autoslice), with the following uncertainties:

$$a_p \in [0.5a \ , 1.5a], \ t_{dp} \in [0.5t_d, \ 1.5t_d], \tag{5.16}$$

where $a = 0.6873$ and $t_d = 2$ for the nominal process, and the corresponding sampling time is $\Delta T = 15s$.

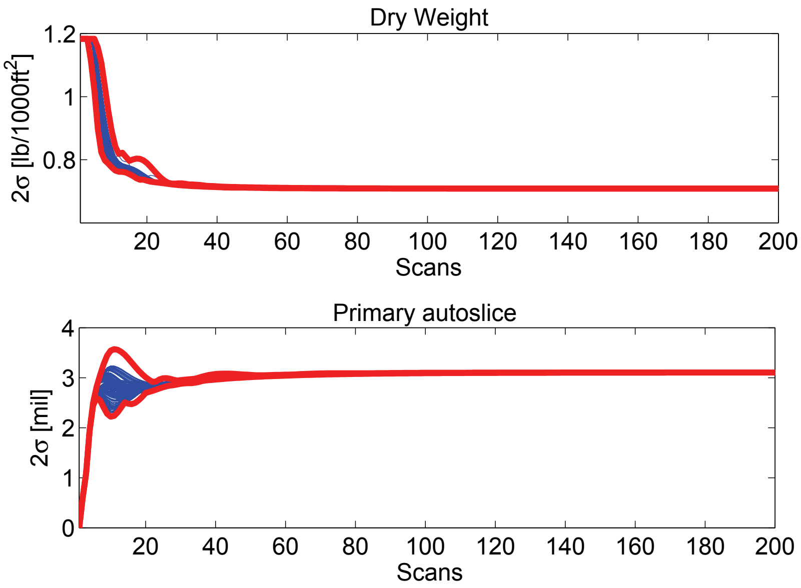


Figure 5.2: Robust performance visualization for the 2σ spreads for the output (upper figure) and the input (lower figure).

5.3.3 An extended robust performance visualization method

The visualization technique can also be extended to characterize the envelope of the 2σ spreads for the best spatial performance under the given parametric uncertainties, by utilizing the technique in Section 4.4.3 to find the best spatial model (with the largest cut-off frequency $v_c = v_l$). Combining the best and worst envelope spreads results in a more detailed visualization result, which is shown in Fig. 5.3. Here the lower red line represents the best 2σ spread (with smallest steady state value and settling time), while the upper red line is for the worst 2σ spread (with largest steady state value and settling time); the area in between indicates all possible 2σ spreads for the given parametric uncertainties; and the blue line represents the nominal case.

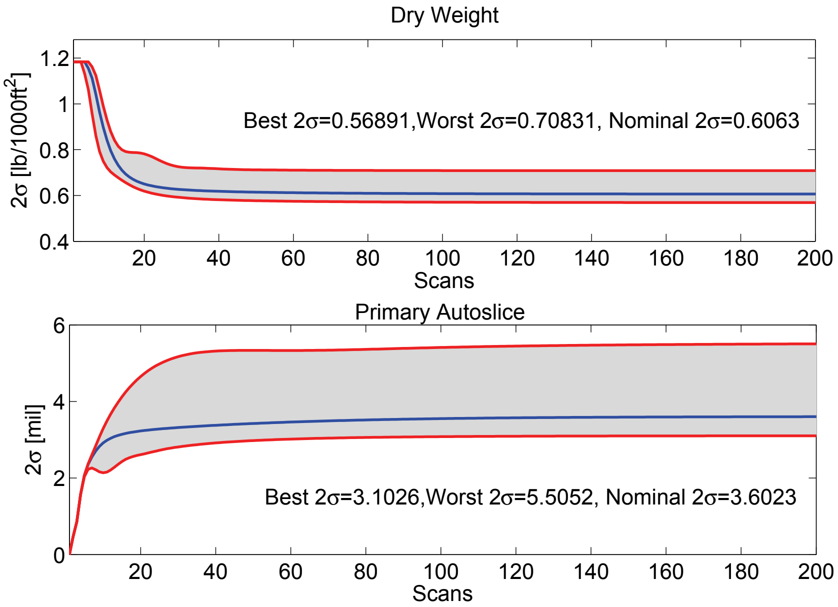


Figure 5.3: Visualization technique for both worst- and best-case 2σ spreads.

5.4 Automated Temporal Tuning Algorithm

Given the proposed performance measures, the tuning objective of the temporal part is then formulated as

$$\begin{aligned} &\min_{\alpha} \mathrm{T}_s(2\sigma(y)) \\ &\text{s.t. } \mathrm{OS}(2\sigma(u)) \leq \mathrm{OS}^*, \end{aligned} \tag{5.17}$$

where $\mathrm{T}_s(2\sigma(y))$ and $\mathrm{OS}(2\sigma(u))$ are the worst-case (longest) settling time of the output 2σ and the worst-case (largest) overshoot of the input 2σ under pre-specified parametric uncertainties, respectively. OS^* is the requirement on the worst-case overshoot. In the above tuning objective, the users can specify the maximum allowable overshoot on the input 2σ, and then solve the optimization problem in (5.17), yielding the tuning parameter that provides the smallest worst-case settling time of the output 2σ. In order to solve the optimization problem in (5.17) efficiently, we separate the temporal tuning into two parts: 1) tuning in the frequency domain; 2) tuning in the time domain.

In the frequency domain tuning, an upper limit on the sensitivity function $T_{yd}(z)$ at $v = 0$ is first selected, and then we find an α to meet the bound while still guaranteeing robust stability.

Define κ_l as the minimum limit of the maximum peak of the sensitivity function (see Fig. 5.4), the objective here is to adjust α so that

$$|\breve{t}_{yd}(0, e^{i\omega})| < \kappa_l, \ \forall \omega, \tag{5.18}$$

is satisfied. Then, the frequency domain tuning can be realized by solving

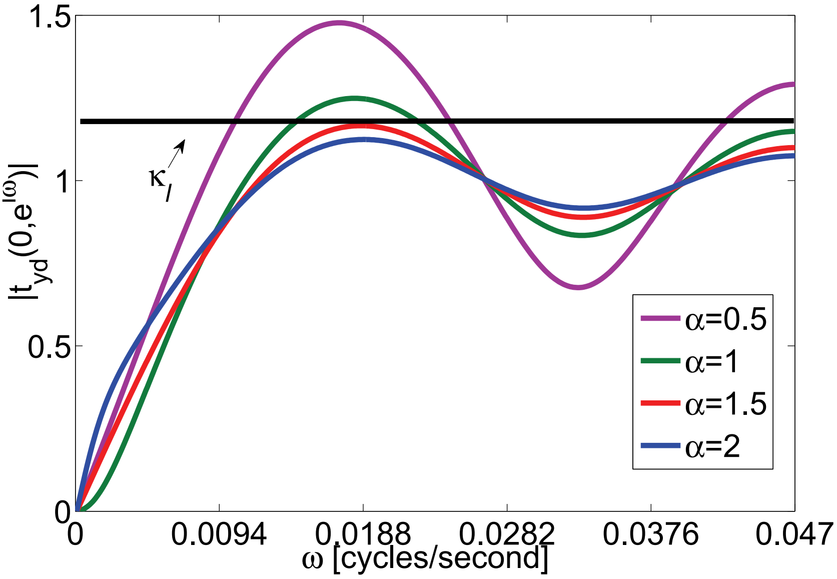


Figure 5.4: The minimum limit on the maximum peak of the sensitivity function.

$$\alpha_f = \max_{k=0,\ldots,k_{max}} (\alpha^m_{\omega_k}) \tag{5.19}$$

$$\alpha^m_{\omega_k} = \min(\alpha_{\omega_k})$$

$$\text{s.t. } \alpha_{\omega_k} > \alpha^*, \ |\breve{t}_{yd}(0, e^{i\omega_k})| < \kappa_l, \tag{5.20}$$

where $\omega_k, \ k = 0, \ldots, k_{max}$, are the temporal frequencies distributed from 0 to the Nyquist frequency [18]. Based on the results in [3] and [18], κ_l can be selected according to OS* and the longest time constant of a process model for the given parametric uncertainties, and is normally between 1.2 and 1.8. It is worth noting that the frequency domain tuning may result in an α with OS $>$ OS*, and thus the time domain tuning is implemented to fine tune the parameter.

In the time domain tuning, the parameter α is fine tuned to find the optimal α (denoted as α^t) that provides the smallest worst-case settling time while still satisfying the requirement on OS. Since OS$(2\sigma(u))$ is a monotonically decreasing function of α, the bisection search can be utilized to find the α^t based on the closed-loop simulation with the following procedure. If the OS with α_f is larger than the specification, the search is implemented in the region $\alpha \in [\alpha_f,\ \alpha_u]$, where α_u is the upper bound for α with $\kappa_l = 1.2$, and α_f is obtained in the frequency tuning. Otherwise, the search is implemented in the region $\alpha \in [\alpha_l,\ \alpha_f]$, where α_l is the lower bound for α with $\kappa_l = 1.8$. To sum up, the temporal tuning part is implemented using Algorithm 11.

Algorithm 11 Temporal tuning

```
Input OS$^*$;
Input the parametric uncertainty specification on temporal model parameters, i.e., $[\underline{a},\ \bar{a}]$ and $[\underline{t}_d,\ \bar{t}_d]$;
Calculate the lower bound for robust stability $\alpha^*$;
Implement the frequency tuning to achieve $\alpha_f$;
if OS$(\alpha_f) >$ OS$^*$ then
    Search in $[\alpha_f,\ \alpha_u]$ for $\alpha^t$;
else
    Search in $[\alpha_l,\ \alpha_f]$ for $\alpha^t$;
end if
Output $\alpha^t$
```

Implementing the tuning procedure in Algorithm 11, the automated temporal tuning normally costs about 6 seconds on a computer with Intel i5 core and 6G memory.

Remark 5.1. *To save more computation time, we can also stop the algorithm if* OS$(\alpha_f) \leq$ OS*. *This can avoid the tuning in the time domain once the requirement is satisfied in the frequency tuning.*

5.5 Simulation Results with Real-Time Simulator

In this section, we apply the proposed automated robust tuning algorithm to a process model obtained from a paper mill in Canada. This is a single beam process in which the dry weight of the output profile is controlled by the primary autoslice.

In this test, the parametric uncertainty specifications are given as

$$\begin{aligned}
&\gamma_p \in [0.75\gamma,\ 1.25\gamma],\ \eta_p \in [0.75\eta,\ 1.25\eta],\\
&\beta_p \in [0.75\beta,\ 1.25\beta],\ \xi_p \in [0.75\xi,\ 1.25\xi],\\
&a_p \in [0.5a\ ,1.5a],\ t_{dp} \in [0.5t_d,\ 1.5t_d],\\
&\epsilon \in [-0.6\frac{m}{n},\ 0.6\frac{m}{n}],
\end{aligned} \tag{5.21}$$

where $\gamma = 0.26434, \eta = 2.3, \xi = 0.3, \beta = 367, m = 264, n = 44, a = 0.6873, t_d = 2$ and the sampling time is $\Delta T = 15s$.

Then, the requirement of the robust temporal tuning of CD-MPC is to guarantee that the time-domain performance indices are satisfied, i.e., a fast settling time with the worst-case overshoot of the actuator 2σ less than $\text{OS}^* = 20\%$.

The overall tuning costs 6.4 seconds on a computer with Intel i5 core and 6G memory. The temporal tuning provides $\alpha = 1.6821$ (see Fig. 5.5, which shows that the requirement $\text{OS}(2\sigma(u)) < \text{OS}^*$ is satisfied), and the spatial parameters are chosen based on the tuning method proposed in Chapter 4. We then apply the obtained tuning results to the Honeywell real time MPC + simulator environment and compare the obtained results with that of the

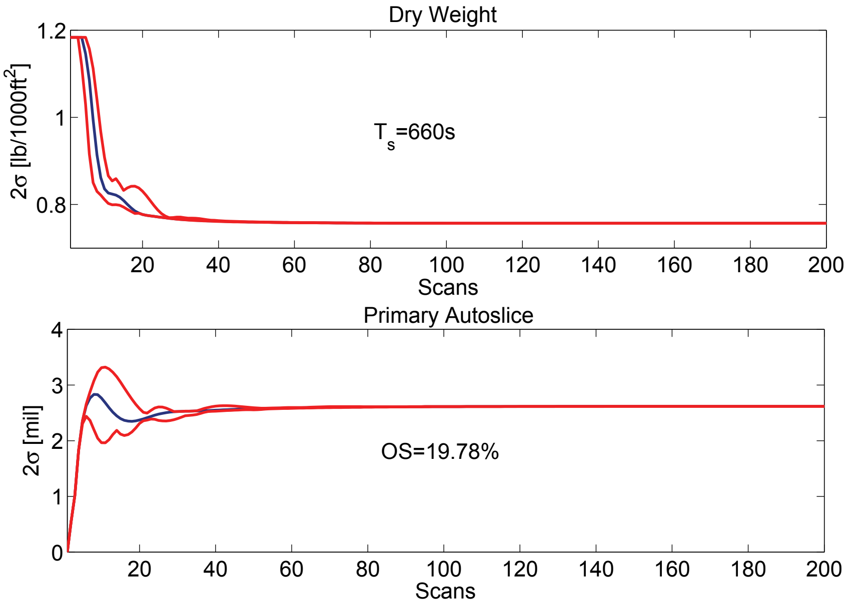


Figure 5.5: Temporal tuning result for the proposed approach. Note that the red lines are the visualization of all the possible 2σs and the blue line is for the nominal model.

existing controller. Note that the real process model G_p is selected satisfying (5.21). The performance is shown in Fig. 5.5, in which the top figure shows the 2σ spread for the measurement profile while the bottom one illustrates the 2σ spread for the corresponding actuator profile. We can see that the proposed tuning approach can achieve much better temporal performance in both the measurement and actuator profiles (2σ settling times are significantly reduced). In addition, it is observed that the steady-state value of the measurement 2σ is even larger than its initial value for the existing controller under the pre-specified parametric uncertainties, which further illustrates the robustness of the proposed method.

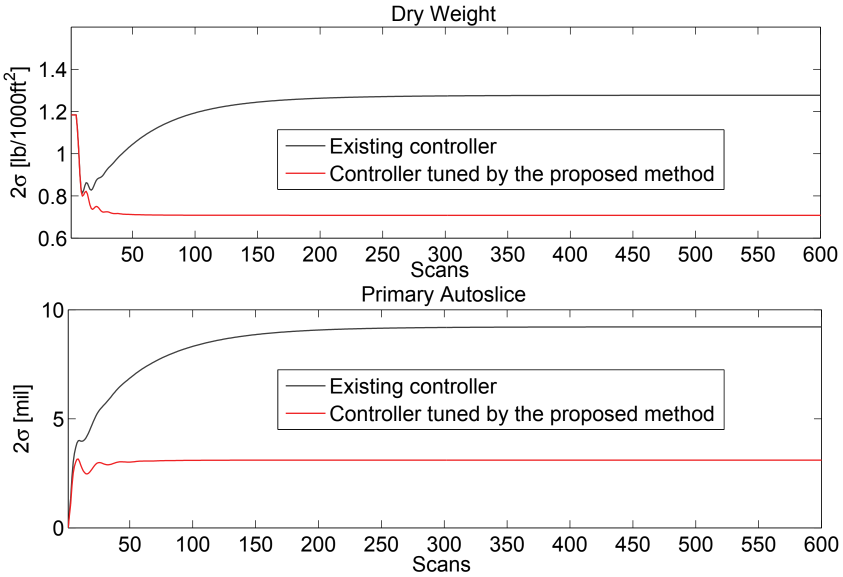


Figure 5.6: Performance illustration for the proposed robust tuning method.

5.6 Summary

The CD-MPC temporal tuning problem for uncertain paper-making processes is explored in this chapter. A visualization technique is first developed to characterize all the possible 2σ spreads given the user-specified parametric uncertainties for CD-MPC. Then, an automatic tuning algorithm is proposed to adjust the parameter in the temporal filter to achieve satisfactory temporal performance as measured by measurement profile 2σ settling times and actuator profile 2σ overshoots. In order to take care of the parametric uncertainties, the tuning is performed based on the worst-case situation, which ensures that the desired performance indices can be achieved. The effectiveness of the proposed algorithm is verified by applying the tuning results on a process extracted from the pulp and paper industry.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, the robust tuning problems of MD and CD MPC have been fully investigated and studied. We have developed a few user-friendly MPC tuning tools for paper-making processes with model parameter uncertainties.

Firstly, the robust MD-MPC tuning problem for uncertain SISO paper-making processes has been explored in Chapter 2. The objective is to achieve satisfactory closed-loop responses, as measured by overshoots, settling times and output oscillations with user-specified parametric uncertainties. As the output variation cannot be easily specified by the end users, two methods are proposed to connect a total variation specification to user-friendly indices, based on which two algorithms are designed to solve the tuning problem. Secondly, the robust MD-MPC tuning problem for uncertain MIMO paper-making processes has been studied in Chapter 3. An envelope algorithm to evaluate the worst-case time domain performance indices considering the interval uncertainty for MD-MPC is first developed, and then a fast MIMO tuning technique is developed to calculated the MPC tuning parameter vectors to fulfill the desired time domain performance under the parametric uncertainty; Besides, a technique to predict the tuning time is proposed to further improve the user-friendliness. Thirdly, the robust spatial tuning problem of CD-MPC has been investigated in Chapter 4. A new weighting matrix S_b is designed to reduce the high-frequency picketing in the actuator profile, and then an automated tuning procedure is developed to adjust the corresponding multipliers such that robust stability and reduced variability of the actuator and measurement profiles are achieved given the user-specified parametric uncertainties.

Lastly, the robust temporal tuning problem of CD-MPC has been studied in Chapter 5. A visualization technique is proposed to evaluate all the possible 2σ spreads given the user-specified parametric uncertainty. A systematic tuning algorithm is then proposed to achieve the user-specified performance, namely, measurement profile 2σ settling times and control signal 2σ overshoots in the presence of parametric uncertainty.

To conclude, the techniques proposed in the PhD thesis provide a solution to the challenge of finding MPC tuning parameters to fulfill user-specified robust performance specifications for paper-making processes with intuitive parametric uncertainties. They have both theoretical and practical significance. The overall concept behind the work may also be relevant to other applications or other control methods where robust tuning is desired; but practitioner capabilities dictate that performance and uncertainty specifications should take simple and intuitive forms.

6.2 Future Work

The research on model predictive control for CD processes, as well as other large scale systems is still at the developing stage and several issues need to be further investigated. In what follows, a number of interesting ones are proposed.

CD-MPC with a hard constraint on the spectrum of the input profile

In Chapter 4, it is mentioned that CD systems are almost uncontrollable above the cut-off frequency v_c, and we have proposed a method to design the weighting matrix S_b such that frequency components beyond v_c in the input profile are penalized by CD-MPC. This works like a soft constraint to limit the magnitude of the input spatial frequency response within the undesired frequency region ($> v_c$). Therefore, it is also desirable to study the possibility of incorporating a hard constraint on the input spectrum of CD-MPC for performance and user-friendliness improvement. The main difficulty for achieving this lies in the construction of such spectrum constraints in the spatial domain. More specifically, the desired spectrum constraints need to be represented in proper forms such that the resultant CD-MPC can be solved by standard QP solvers.

Min-max MPC for CD systems

Given the model-plant mismatch mentioned in the work, the proposed methods focus on adjusting the CD-MPC tuning parameters to achieve robust stability and performance. Another approach to incorporate robustness into CD-MPC is to utilize the a so-called min-max formulation/optimization. The advantage of this approach is the uncertainties on the model parameters can be explicitly considered in the CD-MPC optimization problem and the optimal worst-case performance of the CD process can be guaranteed. However, the min-max formulation introduces additional computational complexity to the CD-MPC optimization problem, and therefore how to reduce the computation time to guarantee the on-line implementation for the modified CD-MPC is the main difficulty in making this idea work.

Event-triggered implementation of MPC for large scale systems

As an optimization based control approach, MPC controllers require a large amount of computation resources for on-line implementation. This restricts its application to only slow dynamic processes (e.g., CD systems considered in Chapters 4 and 5), as the MPC optimization problem needs to be solved at every sampling instant. An event-triggered control strategy is a possible solution for this problem. In such a control strategy, instead of executing the optimization periodically, the computation is triggered only when a prespecified condition is on longer satisfied. This can potentially save more computation and open up the possibility of applying MPC to other large scale processes with relatively fast dynamics. Besides, as shown in [4], the event-triggered control strategy may also provide better performance compared with the original time-based one, and therefore, it is also valuable to investigate the event-triggered MPC for CD processes for performance improvement.

Bibliography

[1] A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafiriou. On-line tuning strategy for model predictive controllers. *Journal of Process Control*, 11(3):265–284, 2001.

[2] D. Angeli, A. Casavola, G. Franzč, and E. Mosca. An ellipsoidal off-line MPC scheme for uncertain polytopic discrete-time systems. *Automatica*, 44(12):3113–3119, 2008.

[3] K. Åström, H. Panagopoulos, and T. Hägglund. Design of PI controllers based on non-convex optimization. *Automatica*, 34(5):585–601, 1998.

[4] K. J. Åström and B. Bernhardsson. Comparison of riemann and lebesque sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, volume 2, pages 2011–2016. IEEE, 2002.

[5] J. Backström, C. Gheorghe, G. E. Stewart, and R. Vyse. Constrained model predictive control for cross directional multi-array processes. *Pulp & Paper Canada*, 102(5):T128–T131, 2001.

[6] J. Backström, B. Henderson, and G. E. Stewart. Identification and multivariable control of supercalenders. *Proceedings of the control systems 2002*, pages 85–91, 2002.

[7] P. Bagheri and A. K. Sedigh. Analytical approach to tuning of model predictive control for first-order plus dead time models. *IET Control Theory & Applications*, 7(14):1806–1817, 2013.

[8] R. A. Bartlett, L. T. Biegler, J. Backström, and V. Gopal. Quadratic programming algorithms for large-scale model predictive control. *Journal of Process Control*, 12(7):775–795, 2002.

[9] C. Bordons and E. F. Camacho. A generalized predictive controller for a wide class of industrial processes. *IEEE Transactions on Control Systems Technology*, 6(3):372–387, 1998.

[10] S. Cairano and A. Bemporad. Model predictive control tuning by controller matching. *IEEE Transactions on Automatic Control*, 55(1):185–190, 2010.

[11] H. Chiou and E. Zafiriou. Frequency domain design of robustly stable constrained model predictive controllers. In *American Control Conference (ACC)*, volume 3, pages 2852–2856, 1994.

[12] D. Chu, M. Forbes, and J. Backström. Technique for converting a model predictive control (MPC) system into an explicit two degrees of freedom (2DOF) control system. U.S. Patent, Application No. 13/907495, 2013.

[13] D. Chu, M. Forbes, J. Backström, C. Gheorghe, and S. Chu. Model predictive control and optimization for papermaking processes. In Tao Zheng, editor, *Advanced Model Predictive Control*, pages 309–341. InTech, 2011.

[14] P. J. Davis. *Circulant Matrices*. Wiley, New York, 1979.

[15] S. R. Duncan. *The cross-directional control of web forming process*. PhD thesis, University of London, UK, 1989.

[16] S. R. Duncan and G. F. Bryant. The spatial bandwidth of cross directional control systems for web processes. *Automatica*, 33(2):139–153, February 1997.

[17] J. Fan. *Model predictive control for multiple cross-directional processes: Analysis, tuning, and implementation*. PhD thesis, University of British Columbia, 2003.

[18] J. Fan and G. E. Stewart. Automated tuning of large-scale multivariable model predictive controllers for spatially-distributed processes. In *American Control Conference (ACC)*, pages 5517–5523, Minneapolis, Minnestota, USA, 2006.

[19] J. Fan, G. .E Stewart, and G. A. Dumont. Two-dimensional frequency analysis and insights for weight selection in cross-directional model predictive control. In *42nd IEEE Conference on Decision and Control (CDC)*, pages 3717–3723, Maui, Hawii, 2003.

[20] J. Fan, G. E. Stewart, and G. A. Dumont. Two-dimensional frequency analysis for unconstrained model predictive control of cross-directional processes. *Automatica*, 40(11):1891–1903, 2004.

[21] J. Fan, G. E. Stewart, G. A. Dumont, J. Backström, and P. He. Approximate steady-state performance prediction of large-scale constrained model predictive control systems. *IEEE Transactions on Control Systems Technology*, 13(6):884–895, 2005.

[22] A. Featherstone, J. VanAntwerp, and R. Braatz. *Identification and control of sheet and film processes.* Springer, Berlin, 2000.

[23] A. P. Featherstone and G. F. Bryant. Control-oriented modeling of sheet and film processes. *AIChE Journal*, 43(8):1988–2001, 1997.

[24] M. Forbes and J. Backström. CD profile reference trajectories for CD-MPC control. Technical Report of Honeywell, 2015.

[25] J. L. Garriga and M. Soroush. Model predictive controller tuning via eigenvalue placement. In *American Control Conference (ACC)*, pages 429–434, 2008.

[26] J. L. Garriga and M. Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.

[27] D. M. Gorinevsky and C. Gheorghe. Identification tool for cross-directional processes. *IEEE Transactions on Control Systems Technology*, 11(5):629–640, 2003.

[28] D. M. Gorinevsky and E. M. Heaven. Performance-optimal applied identification of separable distributed-parameter processes. *IEEE Transactions on Automatic Control*, 46(10):1548–1589, 2001.

[29] D. M. Gorinevsky and G. Stein. Structured uncertainty analysis of spatially distributed paper machine process control. In *American Control Conference (ACC)*, pages 2225–2230, Arlington, VA, USA, 2006.

[30] D. M. Gorinevsky, R. N. Vyse, and E. M. Heaven. Performance analysis of cross-directional process control using multivariable and spectral models. *IEEE Transactions on Control Systems Technology*, 8(7):589–600, 2000.

[31] K. Han, J. Zhao, and J. Qian. A novel robust tuning strategy for model predictive control. In *6th World Congress on Intelligent Control and Automation (WCICA)*, pages 6406–6410, Dalian, P.R. China, 2006.

[32] N. He, X. Liu, M. Forbes, J. Backström, and T. Chen. Robust tuning of cross-directional model predictive controllers for paper-making processes. *IEEE Transactions on Control Systems Technology*. Submitted.

[33] N. He, X. Liu, M. Forbes, J. Backström, and T. Chen. Method of designing model predictive control for cross directional flat sheet manufacturing processes to guarantee spatial robustness and to prevent actuator picketing. U.S. Patent Application No. 15/273,709, 2015.

[34] N. He, X. Liu, M. Forbes, J. Backström, and T. Chen. Method of designing model predictive control for cross directional flat sheet manufacturing processes to guarantee temporal robust stability and performance. U.S. Patent Application No. 15/273,705, 2015.

[35] N. He, D. Shi, M. Forbes, J. Backström, and T. Chen. Method and apparatus for robust tuning of model-based process controllers used with multiple-input, multiple-output (MIMO) processes. U.S. Patent Application No. 14/729,930, 2015.

[36] N. He, D. Shi, M. Forbes, J. Backström, and T. Chen. Robust tuning for machine-directional predictive control of MIMO paper-making processes. *Control Engineering Practice*, 55:1–12, 2016.

[37] N. He, D. Shi, J. Wang, M. Forbes, J. Backström, and T. Chen. Automated two-degree-of-freedom model predictive control tuning. *Industrial & Engineering Chemistry Research*, 54(43):10811–10824, 2015.

[38] J. Hu, C. Bohn, and H. R. Wu. Systematic H_∞ weighting function selection and its application to the real-time control of a vertical take-off aircraft. *Control Engineering Practice*, 8:241–252, 2000.

[39] G. A. Nery Júnior, M. A. F. Martins, and R. Kalid. A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty. *ISA Transactions*, 53(2):560–567, 2014.

[40] H. Kong, G. Goodwin, and M. M. Seron. Predictive metamorphic control. *Automatica*, 49(12):3670–3676, 2013.

[41] D. Laughlin, M. Morari, and R. D. Braatz. Robust performance of cross-directional control systems for web forming processes. *Automatica*, 29(6):1395–1410, June 1993.

[42] W. S. Levine. *The Control Handbook*. Electrical Engineering Handbook. Taylor & Francis, 2010.

[43] C. Mohtadi, S. L. Shah, and D. G. Fisher. Frequency response characterstics of MIMO GPC. *International Journal of Control*, 55(4):877–900, 1992.

[44] N. S. Nise. *Control Systems Engineering*. John Wiley & Sons, Inc., New York, NY, USA, third edition, 2000.

[45] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[46] O. J. Rojas, G. C. Goodwin, and G. V. Johnston. Spatial frequency anti-windup strategy for cross-directional control problems. *IEE Proceedings: Control Theory and Application*, 149(5):414–422, 2002.

[47] C. Rowe and J. Maciejowski. Tuning MPC using H_∞ loop shaping. In *American Control Conference (ACC)*, pages 1332–1336, Chicago, IL, 2000.

[48] G. Shah and S. Engell. Tuning MPC for desired closed-loop performance for SISO systems. In *18th Mediterranean Conference on Control Automation (MED)*, pages 628–633, Marrakech, Morocco, 2010.

[49] G. Shah and S. Engell. Tuning MPC for desired closed-loop performance for MIMO systems. In *American Control Conference (ACC)*, pages 4404–4409, June 2011.

[50] D. Shi and T. Chen. On finite-horizon $l2$-induced norms of discrete-time switched linear systems. *Automatica*, 49(8):2517–2524, 2013.

[51] D. Shi and T. Chen. On finite-horizon optimal control of first-order plus time delay systems. In *27th Chinese Control and Decision Conference (CCDC)*, pages 156–162, 2015.

[52] D. Shi, J. Wang, M. Forbes, J. Backström, and T. Chen. Method and apparatus for specifying and visualizing robust tuning of model-based controllers. U.S. Patent, Publication No. US2015/0268645 A1, Sep. 24 2015.

[53] D. Shi, J. Wang, M. Forbes, J. Backström, and T. Chen. Robust tuning of machine directional predictive control of paper machines. *Industrial & Engineering Chemistry Research*, 54(15):3904–3918, 2015.

[54] D. Shi, J. Wang, and L. Ma. Design of balanced proportional-integral-derivative controllers based on bilevel optimisation. *IET Control Theory & Applications*, 5(1):84–92, 2011.

[55] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, New York, 1996.

[56] G. E. Stewart, D. M. Gorinevsky, and G. A. Dumont. Feedback controller design for a spatially-distributed system: the paper making problem. *IEEE Transactions on Control Systems Technology*, 11(5):612–628, 2003.

[57] G. E. Stewart, D. M. Gorinevsky, and G. A. Dumont. Two-dimensional loop shaping. *Automatica*, 39(5):779–792, 2003.

[58] R. Toro, C. A. Ocampo-Martínez, F. Logist, J. V. Impe, and V. Puig-Cayuela. Tuning of predictive controllers for drinking water networked systems. In *18th IFAC World Congress*, pages 14507–14512, 2011.

[59] Q. N. Tran, L. Özkan, and A. C. P. M. Backx. MPC tuning based on impact of modelling uncertainty on closed-loop performance. In *AIChE Annual Meeting*, Pittsburgh, PA, 2012.

[60] M. Vallerio, V. J. Impe, and F. Logist. Tuning of NMPC controllers via multi-objective optimization. *Computers & Chemical Engineering*, 61:38–50, 2014.

[61] J. G. VanAntwerp, A. P. Featherstone, R. D. Braatz, and B. A. Ogunnaike. Cross-directional control of sheet and film processes. *Automatica*, 43(2):191–211, 2007.

[62] N. Vlassis and R. Jungers. Polytopic uncertainty for linear systems: New and old complexity results. *Systems & Control Letters*, 67:9–13, 2014.

[63] J. Wang, G. Mustafa, T. Chen, and D. Chu. An efficient quadratic programming implementation for cross directional control of large papermaking processes. In *American Control Conference (ACC)*, pages 1073–1078, Montreal, Canada, June 27-29 2012.

[64] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta. Practical approach to tuning MPC. *ISA Transactions*, 42(1):149–162, 2003.

[65] M. Yousefi, B. Gopaluni, P. Loewen, M. Forbes, G. Dumont, and J. Backström. Impact of model plant mismatch on performance of control systems: An application to paper machine control. *Control Engineering Practice*, 43:59–68, 2015.

[66] K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1998.