Design of Multi-parametric NCO-Tracking Controllers for Linear Continuous-time Systems

Muxin Sun

Centre for Process Systems Engineering

Department of Chemical Engineering

Imperial College London

December 2017

This dissertation is submitted for
the degree of Doctor of Philosophy

Declaration

Declaration of Originality

I hereby confirm that the contents of this dissertation are original, and that any ideas or information derived from the published and unpublished work of others have been acknowledged through standard referencing practices of the discipline.

Acknowledgements

First I would like to thanks my supervisor Dr Benoît Chachuat, thanks for everything during my PhD study. Words are powerless to express my gratitude and the reachability tube of my thanks will not converge as time evolves. I would also like to thanks my supervisor Prof Stratos Pistikopoulos, thanks a million from the bottom of my heart for all of your generosity and for your encouragement ever from the first day of my study. Besides, a big appreciation to Dear Mario, not only for your detailed explanations on academic work, but also your supports for encouraging me to overcome the difficulties.

I want to say thanks to the Omega group and the office group, for all the help you gave to me and all of the happy times we spent together. Joy and sadness have all become good memories, and they are valuable treasures in my life. I would like to thanks the staff in the department, for helping me to solve all the problems. I will remember each of you and send my regards from my heart from time to time. And I also want to thanks my Chinese crew, here in London and far away from China, I truly appreciate your cares and supports during my study, and please accept my endless gratitude. To all the people have helped me these years, I would not have become the current me without your help, and I truly want to say thank you.

My gratitude goes to my relatives for all of your cares, and please accept my deepest thanks. Finally, a global maximum love to my family for always being there. It is never enough to say thank you to you all and you are my belief in love.

Abstract

Process optimization for industrial applications aims to achieve performance enhancements while satisfying system constraints. A major challenge for any such method lies in the problem of uncertainty stemming from model mismatch and process disturbances. Classical approaches such as model predictive control usually handle the uncertainty by repeatedly solving the optimization problem on-line, which may prove a rather computationally demanding task nonetheless and cause serious delays for fast dynamic systems. Existing approaches for mitigating the on-line computational burden via off-line optimization include multi-parametric programming and NCO-tracking. Multi-parametric programming aims to generate a mapping of control strategies as a function of given parameters; whereas NCO-tracking involves tracking the necessary conditions of optimality (NCOs) based on a precomputed control switching structure, which enables a dynamic real-time optimization problem to be transferred into an on-line tracking problem using a feedback controller. A methodology, called multi-parametric (mp-)NCO-tracking is developed in this thesis, whereby multi-parametric dynamic optimization and NCO-tracking methods are combined into a unified framework.

An algorithm for the design of mp-NCO-tracking controllers for continuous-time, linear-quadratic optimal control problems is presented in Chapter 2. The off-line step defines the multi-parametric control structure mapped to given uncertain (measurable) parameters in terms of so-called critical regions and feedback laws. Specifically, each critical region corresponds to a unique control switching structure in terms of the sequence of active constraints. The on-line step involves determining the current critical region once the parameter value has been revealed, and then applying the corresponding feedback control laws in a receding horizon manner. The mp-NCO-tracking approach provides a means for relaxing the invariant switching structure assumption in NCO-tracking by constructing critical regions for various switching structures. Moreover, addressing the problem directly in continuous-time can potentially reduce the number of critical regions compared with standard multi-parametric programming based on a time discretization and a control vector parameterization. The methodology and its benefits are illustrated for a number of simple case studies.

To obtain the mathematical representation of the generally nonlinear critical regions, Chapter 3 investigates a machine learning model as a classifier, based on deep neural network. This feed-forward network is selected for its representational power as a universal approximator for arbitrary continuous functions. Here, the classifier takes the unknown parameter as input and maps the corresponding critical regions in terms of their switching structures. An algorithm for training the classifier is presented, which involves generating the training data set, setting up a neural network architecture, and applying optimization based training. By using a Softmax classifier in the output layer of the network, a normalized probability distribution is obtained, which consist of a vector with as many elements as the total number of critical regions, and each element representing the likelihood for a region to be the correct one. The classifier is conveniently embedded into the multi-parametric NCO-tracking controller for choosing the real-time switching structure in on-line control.

Lastly, a robustification of the mp-NCO-tracking methodology is developed in Chapter 4, where constraints are guaranteed to be satisfied under all possible uncertainty scenarios, which leads to a min-max formulation. A robust counterpart formulation of the multi-parametric dynamic optimization problem is presented, which considers both additive or multiplicative time-varying disturbances. The approach involves backing-off the path and terminal constraints of the linear-quadratic optimal control problem based on a worst-case uncertainty propagation computed using either interval or ellipsoidal reachability tubes. The uncertain system state is decomposed into a nominal reference and a perturbed component, and a convex enclosure of the reachable set for the perturbed component is precomputed via some auxiliary differential equations. Conservative constraint back-offs are obtained from the precomputed reachability tubes, which enables the controller design procedure in the nominal case to be directly applied for the robust control problem, and to retain the same computational effort as in the nominal case. These developments are demonstrated by numerical case studies, and ways of extending this approach to more general, nonlinear optimal control problems are discussed in Chapter 5.

Contents

List of Figures

Chapter 1

Introduction

1.1 Dynamic real-time optimization

On-line optimization and real-time control have received much attention over the past few decades because of the need to improve performance and reduce economic costs in industrial processes [1]. Many such industrial applications involve fast dynamic systems operated under constraints, typically reflecting physical operation bounds and/or safety requirements. To obtain the optimal operation, optimization should be applied to make decisions on the best control action without violating certain constraints. The optimal control action can be obtained by solving optimization problems based on dynamic model of the system involving objective function and constraint functions of state and input variables. An illustrative mathematical representation for dynamic optimization problem considered in this thesis with $u(t)$ as control and $x(t)$ as system states can be written as follows:

$$
\begin{aligned}
&\min_{\substack{u(t),x(t),\\ t_0\le t\le t_\mathrm{f}}} \; J(x(t),u(t)) \\
&\text{s.t.} \quad \dot{x}(t) = f(x(t),u(t)) = F_x x(t) + F_u u(t) + f_0 \\
&\qquad\quad g(x(t),u(t)) = G_x x(t) + G_u u(t) + g_0 \le 0 \\
&\qquad\quad h(x(t_\mathrm{f})) = H_x x(t_\mathrm{f}) + h_0 \le 0 \\
&\qquad\quad x(t_0) = x_0 \, ,
\end{aligned}
\tag{1.1}
$$

where t_0 and t_f are the initial and final times; $J(x(t), u(t))$ denotes some performance index that needs to be optimized by choosing the control policy, which is chosen as quadratic objective function $J = \frac{1}{2}x(t_\mathrm{f})^\mathsf{T} Q_\mathrm{f} x(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \frac{1}{2}x(t)^\mathsf{T} Q x(t) + \frac{1}{2}u(t)^\mathsf{T} R u(t)\, dt$ with weighting matrices $Q_\mathrm{f} \succeq 0$, $Q \succeq 0$, and $R \succ 0$; $f(x, u)$ denotes some linear system dynamics with initial condition $x(t_0)$; $g(x(t), u(t))$ and $h(x(t_\mathrm{f}))$ denotes the path and terminal constraints for the process which are linear functions of the state and control variables; and all the other notations denote some constant matrices or vectors. The uncertain parameters considered in this thesis lie in linearly in the dynamics, initial conditions and both path and terminal constraints. Besides, robust control is investigated with additive and multiplicative disturbances in the linear dynamics $f(x(t), u(t))$, such that feasibility is guaranteed for all possible scenarios.

A great variety of algorithms have been investigated for solving dynamic optimization problem, which are generally categorized as indirect methods (variational methods) and the direct methods based on discretization. The former transfers the optimization problem into an infinite-dimensional boundary value problem using the corresponding optimality conditions [2, 3]. In the context of direct approach, the catagary includes simultaneous and sequential methods. The simultaneous approach converts the dynamic optimization problem into a finite dimensional nonlinear program by a discretizaiton of both control and state variables, resulting in a set of algebraic qualities and inequalities, where orthogonal collocation is a popular technique [4–6]. In sequential approach, discretization is achieved via control vector parameterizaiton based on some basis functions that depend on a finite number of decision parameters in the master nonlinear program, which can be subdivided into single shooting and direct multiple shooting methods with continuity constraints for state variables between adjacent time elements [7–9]. The problem with inequality constraints can be solved via either the interior point method [10–12] or sequential quadratic programming [13]. For dynamic optimization, determining global minimum can prove extremely demanding [14] and local optimization techniques are competitive in terms of the computational effort with appropriate variable bounds and good initialization strategies [15].

The standard control scheme in many process industries decomposes a plant's economic optimization into two layers, the real time optimization (RTO) determines optimal plant operation among all feasible steady-states to minimize the economic objective while satisfying all the constraints [16–18], and the advanced control layer tracks the best set point where model predictive control (MPC) is widely employed [19, 20]. The corresponding hierarchical structure from the planning and scheduling layer to process control layer is shown in Figure 1.1.

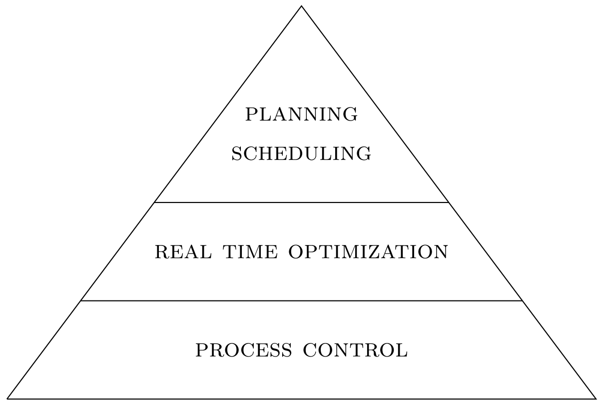


Figure 1.1: Illustration of hierarchical operation

The main drawbacks of a two layer control structure include inconsistent models and economics in the dynamic layer, which may lead the set point for steady state to be unreachable for the dynamic layer [16, 21, 22]. Since the control law designed under hierarchical structure usually overlooks the issue of transient costs, the control actions applied to the dynamic systems may be suboptimal by just steering the system to desired state with minimum transition time, yet without considering the economic performance during transient operations. Instead of splitting the control structure into two layers of equilibrium calculation and set point tracking, dynamic RTO (D-RTO) directly optimizes the economic objective to obtain the optimal control actions [23, 24]. The corresponding structure of dynamic RTO is shown in Figure 1.2, where the two-layers are merged into one centralized decision-making layer, and its the synthesis structure can be formulated as economically-oriented MPC.

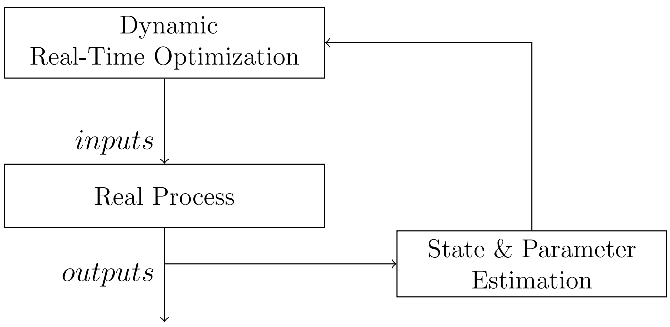


Figure 1.2: An illustration of dynamic-RTO control scheme

A major challenge for process control is the problem of uncertainty stemming from model mismatch and process disturbances where measurement-based method or robust control method needs to be applied. In real applications, uncertainty resulting from model mismatch or disturbances leads the plant to deviate from optimal operation by only implementing the nominal control action. Optimization-based methods that repetitively solve for the optimal control should be applied to optimize system performance while satisfying all the constraints. This can be realized by solving the optimal solutions on-line or resorting to some novel techniques such as multi-parametric programming and NCO-tracking (tracking the necessary conditions of optimality). The rest of the chapter first presents a brief background on model predictive control and some advanced techniques for reducing the on-line computational burdens, including multi-parametric programming and NCO-tracking method. Afterwards, the methodology of the research, multi-parametric NCO-tracking, is introduced alongside a summary of the main contributions, followed by the structure of the thesis.

1.2 Model predictive control

For the current practice in process industry, model predictive control has been widely applied, and the optimal control of industrial plants with stability analysis has reached a solid theoretical foundation [25–27]. Model predictive control, also known as receding horizon control, aims to optimize the system performance in the presence of constraints [16, 28]. It handles the prediction of system dynamic behaviour and constraints at the same time naturally and can accommodate multi-inputs multi-outputs systems as well. Normally, the on-line control implies iteration between parameter estimation and optimization problem solving, while in some real applications, the mathematical modelling used for optimization should be updated on-line via parameter estimation. Economic model predictive control is a recently developed technique for optimizing the economic performance of a system subject to operational constraints [16, 17, 29], which falls into the category of dynamic RTO, where dynamic operation is implemented in a receding horizon manner. Classical MPC stability theory usually assumes a cost function that penalizes deviations from a desired steady state to prove stability of the closed-loop system, while in economic MPC, the cost function may not be strictly decreasing along the closed loop trajectory and the average cost is guaranteed to be no worse than that of the best steady state under specific periodic state-feedback law [30].

Given the current state of the system, an implicit control law $u = \kappa(x)$ can be obtained by repetitively solving an optimization problem online as in the form of (1.1) to compute a virtual optimal control sequence in a predictive time horizon $\mathbf{u}_t = [u_0, u_1, \ldots, u_N]$, as shown in blue dashed lines of input in Figure 1.3 [29, 31]. To compute the optimal control sequence, the dynamic model of the process is used to predict the future dynamic behaviour over a finite time horizon, which might be a couple of minutes or hours, or in infinite time horizon, as shown in blue dashed lines of output in Figure 1.3.

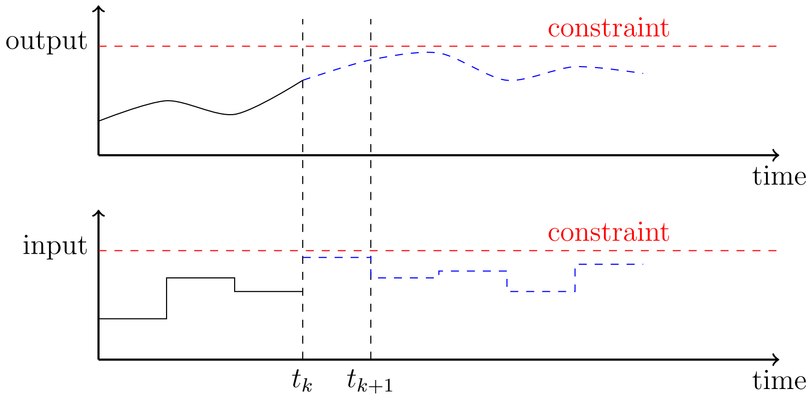


Figure 1.3: MPC: virtual optimal input sequence for t_k

The cost function to be minimized can be defined in the form:

$$V_N(x, u) = f(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i)$$

where $f(x_N)$ and $l(x_i, u_i)$ denote the terminal cost and stage cost, and N denotes the number of time steps for the finite time horizon.

The optimization problem can be written in the following form:

$$\begin{aligned}
&\Phi = \min_{u,x} V_N(x, u) \\
&s.t. \quad x_{i+1} = f(x_i, u_i) \\
&\qquad\quad (x_i, u_i) \in \boldsymbol{Z} \quad \forall i \in \boldsymbol{I}_{0:N-1} \\
&\qquad\quad x_0 = x_t \\
&\qquad\quad x_N \in \boldsymbol{X}_N
\end{aligned}$$

where x_t denotes the initial condition, which is usually obtained from current state feedback; u_i denotes the input sequence as operating variables and x_i denotes the resulting state sequence; $\boldsymbol{Z}$ denotes admissible set for the path and $\boldsymbol{X}_N$ for the terminal constraints to guarantee the feasibility of the system.

After the virtual control sequence is obtained, only the first element is applied to the system for the current time interval, i.e. $\kappa(x_t) = u_0$, as shown in blue solid line in Figure 1.4.

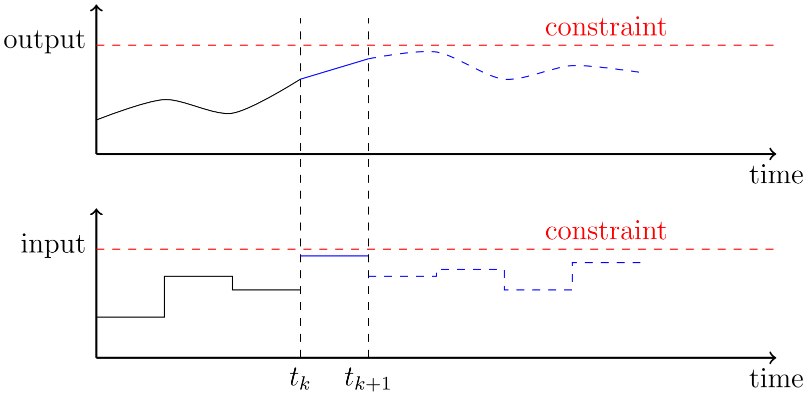


Figure 1.4: MPC: apply optimal input from t_k to t_{k+1}

For the next shifted time horizon starting from t_{k+1}, $x(t_{k+1})$ is taken as the initial condition, i.e. in the following optimization problem, $x_0 = x(t_{k+1})$, and the new virtual control sequence is computed as shown in black dashdotted lines in Figure 1.5.

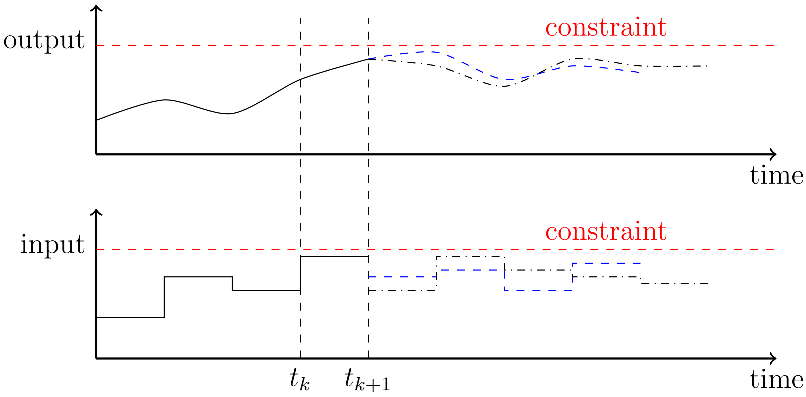


Figure 1.5: MPC: new virtual optimal input sequence for t_{k+1}

1.3 Towards off-line computations

The strategy employed in MPC entails the repeated solution of an optimal control problem that predicts the system's future behaviour over a finite, receding time-horizon, using the current state measurements or estimates as initial conditions [32]. The optimized control strategy is implemented until the next measurements become available, and it is the repetition of this process that creates a feedback control. This may prove to be a rather computationally demanding task nonetheless and cause serious delays for fast dynamic systems, thereby leading to performance deterioration or even infeasibility [33, 34].

The computational burden associated to the on-line solution of optimization problems in MPC could be mitigated by solving optimization problems off-line, such as using multi-parametric programming and NCO-tracking, which are introduced in this section. The former method computes the optimal input in an explicit scheme by solving optimization problem off-line [35–37]. For the latter case, instead of solving optimization problem on-line, optimal input is updated directly from system feedback by tracking necessary conditions of optimality corresponding to the optimization problem [38–41].

In the multi-parametric programming paradigm [42–44], solving optimization problem is performed off-line, resulting in an explicit mapping of the control strategies as a function of the initial state of system. For continuous-time systems, this approach gives rise to multi-parametric dynamic optimization (mp-DO) problems, which may either be transformed into finite-dimensional multi-parametric programs via full discretization (direct approach) or handled directly using optimal control theory (indirect approach). In addition, the explicit solution can be applied in MPC, which is referred to as mp-MPC or explicit MPC, and an indicative list of key publications is given in Table 1.1.

Table 1.1: Publications on multi-parametric/explicit model predictive control

Multi-parametric MPC	Pistikopoulos (1997, 2000), Pistikopoulos and Morari (2002), Bemporad et al. (2002), Johansen and Grancharova (2003)
Multi-parametric continuous time MPC	Kojima and Morari (2004), Sakizlis et al. (2005)
Hybrid multi-parametric MPC	Bemporad et al. (2000), Sakizlis et al. (2001, 2003), Borrelli et al. (2005)
Robust multi-parametric MPC	Bemporad et al. (2001), Kakalis (2001), Sakizlis et al. (2004), Mayne et al. (2006)
Multi-parametric nonlinear MPC	Johansen (2002), Bemporad (2003), Sakizlis et al. (2007), Dominguez and Pistikopoulos (2010 ,2012), Ziogou et al. (2013)

Besides, a data-driven framework for mp-MPC was proposed based on surrogate modelling and data-classification [45], where the parameter space is divided into regions by using Support Vector Machine [46, 47], and in each region, the first input value of mp-MPC is expressed as an approximate function of states. Thus, the on-line control problem can be computed efficiently by substituting the explicit pre-computed solution mapping for on-line repetitively optimization. Comparison of the closed-loop control profile by directly solving optimization problem on-line and solving off-line using multi-parametric programming is shown in Figure 1.6 (a)(b), where blocks linked with dashed lines denote off-line calculation and blocks linked with solid lines denote on-line updating. The multi-parametric programming approach provides solutions as piecewise affine functions of the parameters, but the resulting solution map usually consists of very large number of critical regions. With the increase on the scale of optimization problems or constraints, the number of critical regions increases significantly.

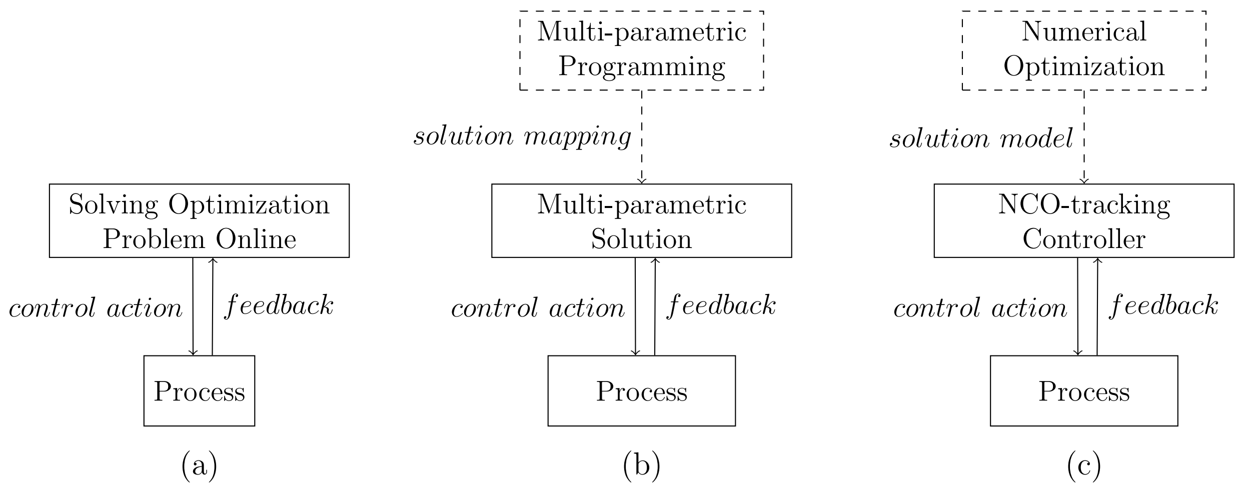


Figure 1.6: Comparison between solving on-line and off-line

Another approach to reduce the computational burden in MPC involves tracking the necessary conditions for optimality (NCO), namely *NCO-tracking* [40, 48]. In continuous-time, NCO-tracking starts by characterizing the optimal switching structure of the control trajectories. Under the assumption that this switching structure remains unchanged in the presence of uncertainty, feedback laws are then constructed for tracking the active constraints and zero gradient conditions along each arc, and the comparison with on-line optimization is shown in Figure 1.6 (a)(c). A key limitation with the current NCO-tracking methodology nonetheless lies in the fact that the underlying optimal control switching structure might change in the presence of uncertainty; thus, this research aims to close the gap between multi-parametric and NCO-tracking approach for enabling changing switching structures.

1.4 Contribution and overview

This thesis presents a methodology for combining multi-parametric programming and NCO-tracking into a unified framework, for which we coin the name *multi-parametric NCO-tracking control*. By taking advantage of both methods, dynamic real-time optimization is transferred into an on-line tracking problem using feedback controller. Such a combination is especially promising in that multi-parametric programming provides a means for relaxing the fixed switching structure assumption in NCO-tracking, thereby paving the way towards a theoretical justification for NCO-tracking. In addition, the use of feedback laws tracking the optimality conditions inside multi-parametric controller could provide a means for reducing the number of critical regions compared with classical mp-MPC controllers based on control vector parameterization, where the critical region for each multi-parametric switching structure can be regarded as an union of all critical regions in the discrete time case sharing the same switching structures, i.e. the same sequence of active constraints.

In Chapter 2, an algorithm for characterizing the corresponding multi-parametric solution structure in terms of the exact critical regions and nonlinear feedback control laws was proposed for linear-quadratic optimal control problems. A review for both multi-parametric programming and NCO-tracking is first presented in Chapter 2, followed by the detailed presentation of the proposed mp-NCO-tracking control based on the necessary conditions of optimality. By solving the mp-DO problem for continuous-time linear dynamic system, the feasible regions, which might be characterized by the initial conditions of the systems or some model parameters, are divided into different critical regions. Each region corresponds to an unique control switching structure, i.e. the sequence of arcs, with regard to either active constraints or sensitivity conditions.

The on-line step by using the parametric controller involves determining the critical regions containing the measured parameter and applying the corresponding feedback law until next measurement is available. This control framework is readily applied in the receding horizon control. Estimates of states and model parameters can be obtained via on-line estimation approach, enabling the solution structure to be updated by locating the parameter into the corresponding critical region. At the end of the chapter, the framework is illustrated in several case studies for both mp-DO solutions and closed-loop simulations, where parameter jumps between different critical regions.

For implementation of mp-NCO-tracking control, a potential difficulty exists in characterization of the critical regions, where explicit expressions can not be obtained in general. This is due to the fact that the critical regions in mp-DO may be non-convex and closed-form expressions describing their boundaries may not be avaialable in general. Based on fast-developing field of machine learning for classification, data-driven method is applied for the characterization of critical regions in Chapter 3. Among the classification methods in machine learning, neural network is chosen as the model for critical region classification for its ability to represent non-linearities of arbitrary complexity.

The algorithm for training a classifier comprises setting up a neural network model, generating sample data and performing optimization of the network off-line. Once the training process is completed, training data can be completely discarded, and the classifier is embedded into the mp-NCO-tracking controllers. The architecture of fully-connected neural network model is presented followed by training the model, where optimization is used for choosing the parameters that characterize the network. Afterwards, an algorithm for the implementation of mp-NCO-tracking is presented, from model set up to trained classifier. Finally, the developments are illustrated by case studies at the end of the Chapter 3.

Both classical and multi-parametric MPC controllers are often implemented without considering external disturbances or model mismatch. When large disturbances occur, the constraints can nonetheless become violated, thereby calling for the development of robust control schemes. Thus, a robustification of the mp-NCO-tracking approach is presented in Chapter 4 by extending the multi-parametric controllers for continuous-time linear dynamic systems subject to bounded uncertainty. The approach involves backing-off the path and terminal constraints based on a worst-case uncertainty propagation, and it is computed based on either interval or ellipsoidal reachability tubes. The uncertain state is decomposed into a nominal reference and a disturbed component, and the effect of disturbance is incorporated into the dynamics of the disturbed component, enclosed by a reachability tube which is centred at the nominal reference trajectory.

A robust-counterpart formulation of the mp-DO problem is formulated considering both additive and multiplicative time-varying uncertainty. Here, the robust formulations retain tractability of the mp-NCO-tracking design problem, thus enabling the direct application of the controller design procedure as in the nominal case and making the off-line computational effort independent of the number of uncertain parameters. The result of backing-off the constraints is a modification of the size of the critical regions and possibly the number of critical regions as well—either removing or adding extra regions. The applicability of the approach is demonstrated by case studies for closed-loop simulations.

1.5 Structure of the thesis

The rest of the thesis is organized as follows: Chapter 2 presents the multi-parametric NCO-tracking methodology based on solving the mp-DO problem and the control algorithm; Chapter 3 focuses on training a classifier for characterization of critical regions corresponding to the multi-parametric switching structures used for point location embedded in the mp-NCO-tracking controller; Chapter 4 develops robust mp-NCO-tracking control based on a robust-counterpart formulation of the mp-DO problem for both additive and multiplicative time-varying uncertainty, considering worst-case scenarios based on set-propagation techniques; Finally, Chapter 5 concludes the thesis and discusses the future directions.

Chapter 2

Multi-parametric NCO-tracking control for linear dynamic system

2.1 Introduction

Optimization has played a crucial role in process control over the past years. Optimal control strategies can be determined by solving constrained optimization problems based on a dynamic model of the system. One major challenge with this approach is how to effectively manage uncertainty stemming from model mismatch and process disturbances, as optimal operation needs to be decided on-line using real-time feedback. The strategy employed in classical model predictive control (MPC) [32] handles this uncertainty by repeatedly solving the optimization problem on-line in order to update the optimal inputs. This is often a rather computationally demanding task that may cause serious delays especially for systems with fast dynamics, leading to suboptimal performance or even infeasibility.

In the multi-parametric programming paradigm, the optimization is performed off-line, resulting in a priori explicit mapping of the solutions, effectively control strategies, as a function of measurable quantities [42]. For continuous-time systems, this approach calls for the solution of multi-parametric dynamic optimization problems (mp-DO) [49]. Another approach to reducing the on-line computational burden involves tracking the necessary conditions for optimality, namely NCO-tracking [40, 41, 48]. There, an optimal control policy is obtained indirectly by forcing the NCOs to zero. This process requires knowledge of the switching structure of the optimal control, based on which feedback control laws can be derived on account of the available output measurements, effectively converting an optimal control problem into a set of self-optimizing feedback control laws [40, 50]. However, a key assumption for this controller to enable optimal or even feasible operation is that the switching structure should remain unchanged, which may not be the case when large uncertainty is present.

This chapter presents a methodology for combining mp-DO and NCO-tracking into a unified framework for model predictive control, for which we coin the name *multi-parametric NCO-tracking control*. Such a combination is especially promising in that multi-parametric programming provides a means for relaxing the fixed switching structure assumption in NCO-tracking, while the use of feedback laws by NCO-tracking enables the number of critical regions compared to classical mp-MPC controllers based on control vector parameterization to be dramatically reduced.

The rest of the chapter is organized as follows. Section 2.2 provides some background for both multi-parametric programming and NCO-tracking method. Section 2.3.2 presents the multi-parametric NCO-tracking methodology, including the use of mp-DO for mapping subregions of the uncertain parameters to various switching structure and the implementation of the corresponding NCO-tracking controllers in a receding horizon manner. Several numerical examples are given in Section 2.4 to illustrate the approach. Finally, Section 2.5 concludes the paper.

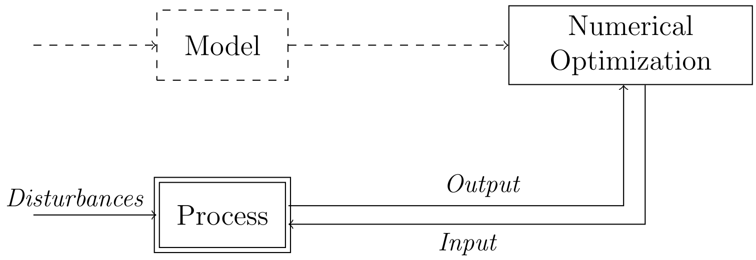
2.2 Background

2.2.1 Review of multi-parametric programming

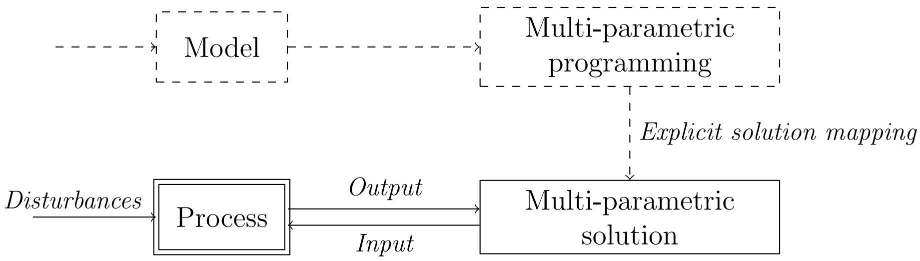
Multi-parametric (mp) programming is a technology that allows determining the optimal solution as a function of parameters [37], which provides a means for computing the solution mapping of an optimization-based control problem off-line based on a model of the dynamic system. The optimal control trajectory is expressed as a function of given parameters, usually some uncertain measured quantities, thus avoiding the need for repeatedly solving optimization problems on-line when these parameters vary [35–37, 43, 44]. The multi-parametric solution calls for the solution of multi-parametric dynamic optimization problems (mp-DO) for continuous-time systems. In practice, such mp-DO can either be transformed into a finite-dimensional multi-parametric program via control vector parameterization [49] or handled directly into its native infinite-dimensional form using the corresponding optimality conditions. In the context of solving the finite-dimensional multi-parametric program, numerous publications exist [35, 37, 42, 51–53], whereas solving the infinite-dimensional counterpart has received relatively little attention [49]. The continuous-time solutions ask for nonlinear critical regions as opposed to its discrete-time counterpart. Thus, the standard techniques for generating linear critical regions can not be applied, and multi-parametric solution can hardly be obtained in general without considering the switching structures, which resort to the concept of NCO-tracking. The general theory and algorithms of multi-parametric programming and its applications has been studied in the books [43, 44], and publications in various algorithms of multi-parametric programming are well summarized in [37, 51]. Overall, the procedure for obtaining multi-parametric solution includes generating critical regions and piecewise affine functions. However, the detailed algorithms may vary based on different types of the optimization problem, which might be linear or nonlinear, with only continuous variables or with integer variables.

Parametric optimal solutions for the infinite-dimensional problem can be obtained by sensitivity analysis, also known as neighbouring extremal (NE) control. In this approach, a feedback law is derived in the neighbourhood of a nominal solution, where the switching structure—namely, the sequence of active path and terminal constraints—remains the same [54–57]; see also [55, 58, 59] for a discussion of the differentiability and stability of parametric solutions. For constrained linear-quadratic optimal control problems in particular, Sakizlis et al. [49] have shown that the mp-DO can be written as a multi-parametric boundary value problem using the Pontryagin Minimum Principle [2, 60, 61]. The continuous-time optimal control trajectory is expressed as a time-varying function of selected parameters, which provides a means for determining the control switching structure using standard multi-parametric programming techniques.

In the context of MPC, multi-parametric programming can be used to obtain the optimal solution of control action as explicit functions of system state feedback or other estimated parameters [37, 42]. A comparison of the control framework of MPC and mp-MPC is shown in Fig. 2.1. Instead of repeatedly solving the optimization problem during run-time as in MPC, mp-MPC computes a mapping between the uncertain parameters and their corresponding optimal solutions off-line, and then simply selects the pre-computed control law at run-time after the uncertainty is revealed. The main steps for generating the explicit parametric controller is as follows [42]: develop a high fidelity model of real system and make suitable approximation by system identification or model reduction; design a robust explicit controller; implementation and validation of the designed controller.



(a) (classical) MPC

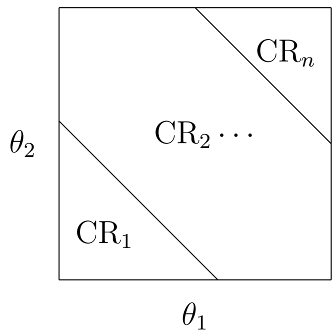


(b) mp-MPC

Figure 2.1: Framework of receding horizon control. Dashed lines: off-line task. Solid lines: on-line task.

In the multi-parametric solution based on control vector parameterization, the parameter space is partitioned into a number of critical regions, and the optimal input variable, u, is described by a given function of the parameters, θ, as

$$u(\theta) = \begin{cases} \kappa_1(\theta) & \text{if } \theta \in \mathrm{CR}_1, \\ \kappa_2(\theta) & \text{if } \theta \in \mathrm{CR}_2, \\ \vdots & \\ \kappa_n(\theta) & \text{if } \theta \in \mathrm{CR}_n. \end{cases} \tag{2.1}$$



with the critical region illustrated by a two-dimensional parameter space. Here, $u(\theta)$ is a finite-dimensional vector that only depends on θ in each critical regions. Each critical region corresponds to a unique combination of active constraints and an uniquely defined control law. The boundary of the critical region can be computed from sensitivity analysis of the KKT conditions by keeping the inactive constraints non-binding and the multipliers associated with the active constraints non-negative.

A basic procedure for determining the critical regions is the following [37]:

0. Define the uncertainty domain Θ, and set $N = 0$.
1. Select a feasible point θ in the region $\Theta \setminus \cup_{i=1}^{N}\mathrm{CR_i}$. If no such point exists, stop; else, set $N \leftarrow N + 1$.
2. Construct the critical region $\mathrm{CR_N}$ around θ, wherein the active constraints are the same, e.g. using sensitivity analysis of the KKT conditions.
3. Return to step 1.
4. Unify the regions and solutions for a more compact representation.

On termination, this procedure returns the number N of critical regions contained in the initial domain Θ. In step 2 for the critical region construction, let $\breve{g}(u, \theta)$ denote inactive constraints and $\hat{\lambda}(u, \theta)$ denotes the Lagrangian associated with active constraints; then the critical regions including the feasible point selected can be defined as:

$$\mathrm{CR_N} := \begin{cases} \breve{g}(u, \theta) \leq 0 \\ \hat{\lambda}(u, \theta) \geq 0 \end{cases}$$

For linear problems involving multi-parametric linear programming (mp-LP) and multi-parametric quadratic programming (mp-QP), there exist exact solutions [35], and details about formulating the problems can be found in [62]. The main result of multi-parametric programming is the solution mapping of the control problem, where the input can be expressed as a linear function of parameters $u = a_1\theta + a_0$. A linear mp-MPC problem is presented later in the following paragraphs to illustrate the procedure based on solving a mp-QP problem (2.3). The same example is used throughout the theoretical part of this chapter to illustrate the developments.

Illustrative example We consider the problem to steer the state $x(t)$ of the following second-order system to zero, by manipulating the bounded input $u(t) \in [-2, 2]$:

$$\dot{x}(t) = \underbrace{\begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}}_{F_x} x(t) + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{F_u} u(t)\,. \tag{2.2}$$

The mp-MPC problem obtained by discretizing the dynamics on N time intervals along the time horizon $0 \leq t \leq 1$ reads

$$\begin{aligned} \min_{u,x}\ & x_N^\mathsf{T} Q_\mathrm{f} x_N + \frac{1}{N} \sum_{k=0}^{N-1} x_k^\mathsf{T} Q x_k + u_k^2 \\ \text{s.t.}\quad & x_{k+1} = \overline{F}_x x_k + \overline{F}_u u_k, \quad k = 0 \dots N-1 \\ & -2 \leq u_k \leq 2, \quad k = 0 \dots N-1 \\ & x_0 = \theta, \end{aligned} \tag{2.3}$$

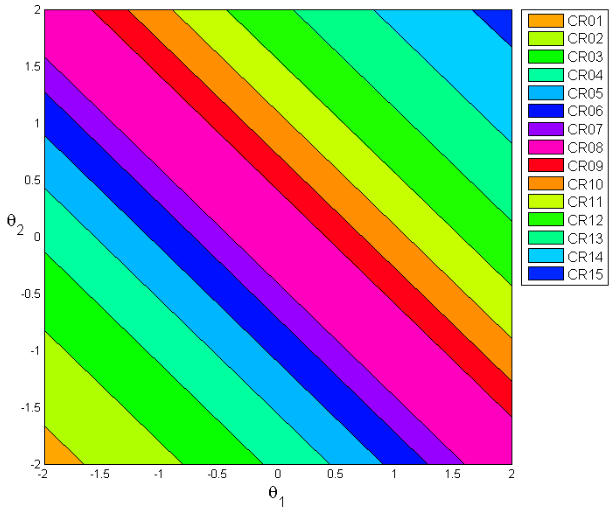
where the parameters $\theta \in [-2, 2]^2$ corresponds to the initial state; the matrices $\overline{F}_x$ and $\overline{F}_u$ in the discretized system are given by

$$\overline{F}_x = \exp\left(F_x T\right) \quad \text{and} \quad \overline{F}_u = \left[\int_0^T \exp\left(F_x(T-t)\right) dt\right] F_u\,,$$

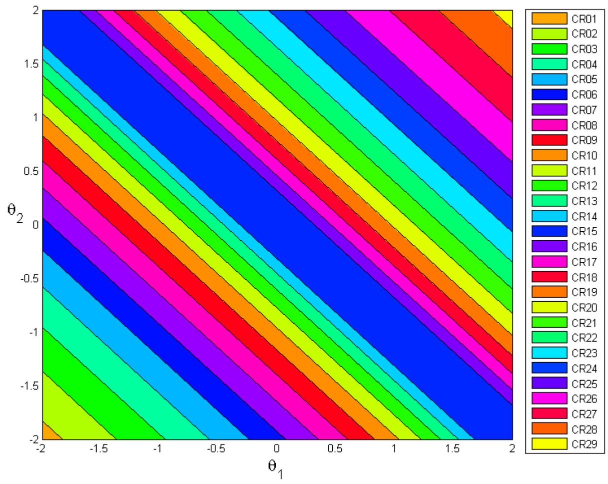
with the sampling time $T := 1/N$; and the weighting used in the objective function is as follows

$$Q_\mathrm{f} = \begin{bmatrix} 0.8198 & 0.8198 \\ 0.8198 & 10.82 \end{bmatrix}, \quad Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \text{ and } \quad R = 0.1\,.$$

Numerical solution of the mp-QP (2.3), here using the PAROC framework [52], provides expressions of the optimal controls $u = [u_0, \dots, u_{N-1}]$ as explicit functions of the initial conditions θ, in the form (2.1). The critical regions for the optimal solution are shown in Fig. 2.2(a) in the case $N = 10$, whereby each region CR_i corresponds to a piecewise affine functions $u = K_i\theta + k_i$, with $K_i \in \mathbb{R}^{N\times 2}$ and $k_i \in \mathbb{R}^N$. Here, the region labelled CR08 in the central part corresponds to the case that none of the input constraints are active. The regions above CR08 correspond to the input lower bound being active for one or more time intervals, and the number of active constraints increases as the distance from CR08 increases. Likewise, the regions below CR08 correspond to the input upper bound being active for one or more time intervals.



(a) $N = 10$



(b) $N = 20$

Figure 2.2: Critical regions for the mp-QP (2.3).

For comparison, critical regions in the case $N = 20$ are shown in Fig. 2.2(b). Here again, the region in the center corresponds to unconstrained solution and other regions, either above or below it, correspond to input constraints being active for the first few time intervals. The multi-parametric solution becomes more accurate due to the use of a smaller sampling time, but at the same time the number of critical regions increases significantly, thereby defining a trade-off between accuracy and computational tractability. In contrast, the approach proposed in this paper removes the need for discretizing the dynamics and the control trajectories, in order to reduce the number of critical regions.

2.2.2 Review of NCO-tracking

NCO-tracking is a measurement-based optimization approach to enforce optimality in the presence of uncertainty via tracking of the necessary conditions for optimality (NCO) [40, 63]. This way, a dynamic optimization problem is transformed into a feedback control problem, which may lead to a large reduction of the on-line computational effort by avoiding the repeated solution of an optimal control problem.

The design of the NCO-tracking controllers starts by detecting the switching structure of the optimal control in order to formulate a feedback strategy via appropriate pairing between the input and output variables—the so-called *solution model* [39, 41], see Fig. 2.5(a). In the feedback loop, input is updated directly by enforcing the system to meet the corresponding necessary conditions in solution model. A specific switching structure is characterized by a unique sequence of arcs and the solution model involves the types of each arc and the switching times [39], which normally comes from the concept of minimum principle. For the solution of a dynamic optimization problem, the optimal trajectory corresponds to a unique sequence of active path constraints and active terminal constraints, which can be characterized by solving the first-order NCO for the problem. More details can be found in [2, 39, 64], and the corresponding equations of optimal conditions for the dynamic optimization problem considered in this thesis consists of (2.6) - (2.19) in section 2.3.2.1.

An illustration of a sequence of arcs and switching times in solution structure can be found in Figure 2.3, where u_1, u_2, u_3 denote different arcs that specific optimal conditions need to be satisfied between time intervals defined by switching times t_1, t_2. The switching times as well as the parameters in parameterized input profiles are the decision variables to be adjusted through NCO-tracking.

Along each arc, a certain combination of inputs may be used for tracking the active path constraints, whereas the remaining inputs are adapted for forcing stationary conditions (gradients) to zero. This latter forcing usually calls for approximation techniques, such as neighboring-extremal control [38, 65–67] or extremum-seeking control [68, 69]. The sensitivities of the control variables with respect to the uncertain parameters are obtained through variation of the first-order NCO at the nominal solution [70, 71]. For singular problems, by the singular value decomposition to split input profiles into nonsingular and

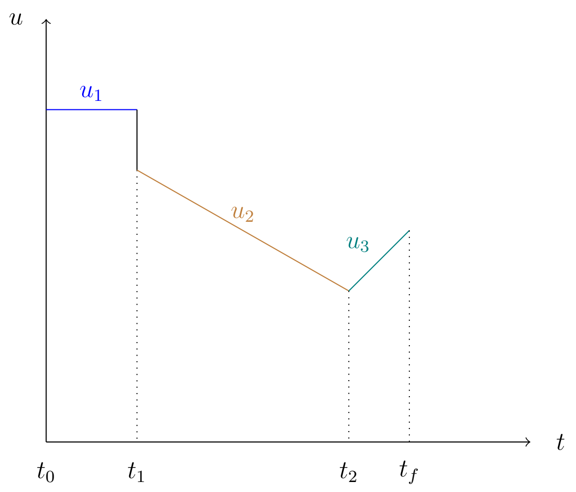


Figure 2.3: A switching structure with three arcs

singular parts, the latter can be determined form additional differentiation [72]. Combining NCO-tracking and self-optimizing together is discussed in [73], where both represent first-order approximations to the classical two-step approach of RTO [74]. It is sometimes possible to arrive at a fully decentralized control scheme, for instance using directional information [40, 50, 75]. In order to steer process to the optimal operation, the online feedback controller requires reliable estimates of NCO parts which are possible for path constraints but not for terminal constraints or sensitivity arc, where model based prediction is needed for making decision.

It may be noted that the sequence of arcs in solution model actually contains the same information as the critical region in the context of multi-parametric programming. The unique combination of active constraints in multi-parametric programming also corresponds to a sequence of active constraints in the time domain, which can be considered as a certain switching structure with several arcs.

An example of emulsion copolymerization of styrene/α-methylstyrene in a batch reactor [50] is presented here to explain the solution structure as a sequence of arcs. The reactor temperature $T(t)$ is manipulated to minimize the batch time while satisfying the bounded constraints on reactor and jacket inlet temperature as well as some performance demand.

The optimization problem based on the dynamic model in [76] for this case is as follows

$$
\begin{aligned}
\min_{T(t),t_f} \quad & t_f \\
s.t. \quad & \dot{x}(t) = f(x(t), u(t)) \\
& x(0) = x_0 \\
& T_{min} \leq T(t) \leq T_{max} \\
& T_{jin_{min}} \leq T_{jin}(t) \\
& X(t_f) \geq X_{f_d} \\
& M_n(t_f) \geq M_{nf_d}
\end{aligned}
$$

where the first constraint is the tendency model for the process, including state variables like monomer concentration and reactor and jacket temperature. The following constraints correspond to bounds on the reactor temperature, the jacket inlet temperature, the final conversion and number average molecular weight, respectively. The optimal operation of reactor temperature, as shown in Figure 2.4, consists of a sequence of four arcs, each corresponds to some active constrains or sensitivities conditions: arc 1 – reactor temperature is maintained at its maximal value T_{max} to accelerate the nucleation, i.e. the third constraint of its upper bound is active; arc 2 – reactor temperature takes value between T_{max} and T_{min} to avoid shorter polymer chains and lower molecular weights by imposing lowest jacket inlet temperature $T_{jin_{min}}$, i.e. the four constraint on its lower bound is active; arc 3 – temperature is maintained at some optimal constant value, for any increase will lead to smaller average molecular weight; finally arc 4 – the last step without monomer supply and the reactor temperature should be increased to compensate for a drop in reaction rate with the increasing slope α as a decision variable, with the initial temperature fixed at the temperature of arc 3. Besides, the two terminal constraints are forced to be active for the batch to be optimal. In total, there are three jumps between arcs at switching times t_1, t_2 and t_3, where optimal control policy switches to meet the specific operating conditions corresponding to each arc.

A key limitation with the current NCO-tracking methodology nonetheless lies in the fact that the underlying optimal control switching structure is known in advance and assumed to keep unchanged, but it might change in the presence of uncertainty. As a result, the NCO-tracking controller may be suboptimal and could even lead to infeasible operation due to constraint violation. Although the assumption of a constant structure is often

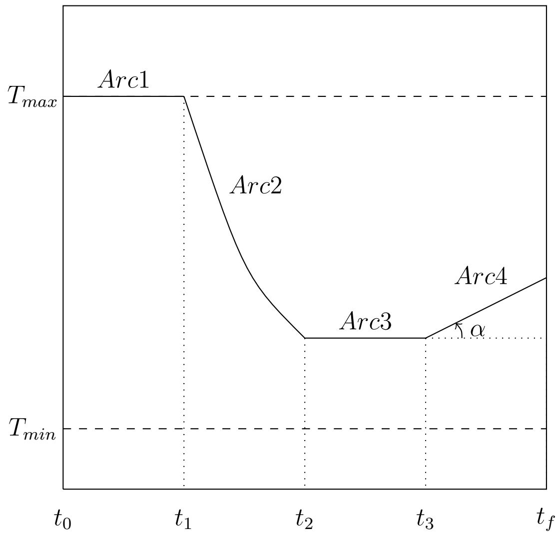


Figure 2.4: Nominal optimal reactor temperature profile

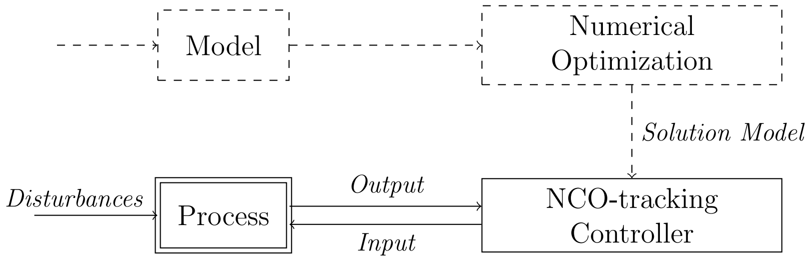
satisfied in batch process optimization applications [77], it is not well suited for MPC applications where constraints frequently activate or deactivate.

It has been suggested that the control switching structure could be monitored by some supervisory system [38]. The developments in this chapter provide the foundations for such an approach to handling a varying optimal switching structure, such that process operation remains feasible and optimal. The operation relies on the mapping between uncertain parameters and optimal switching structures using mp-DO, and the subsequent formulation of optimal control laws that can be applied in a receding horizon manner, namely multi-parametric NCO-tracking controllers.

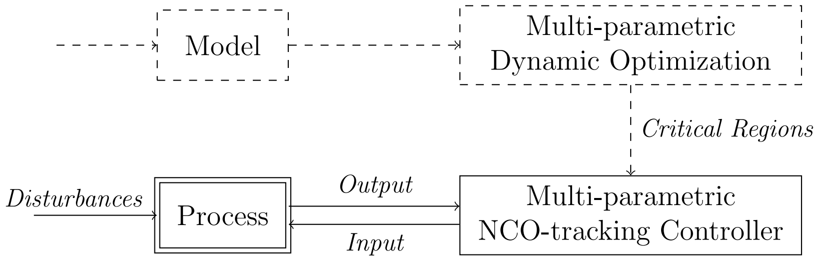
2.3 Methodology for multi-parametric NCO-tracking control

2.3.1 Problem statement

The main contribution of this chapter is a methodology for the derivation of multi-parametric NCO-tracking controllers for constrained linear-quadratic optimal control prob-



(a) classical NCO-tracking



(b) multi-parametric NCO-tracking

Figure 2.5: Principle of NCO-tracking methodology

lem in the form:

$$\begin{aligned}
\Phi(\theta) \quad &:= \min_{\substack{u(t),x(t),\\ t_0\le t\le t_\mathrm{f}}} \frac{1}{2}x(t_\mathrm{f})^\mathsf{T}Q_\mathrm{f}x(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \frac{1}{2}x(t)^\mathsf{T}Qx(t) + \frac{1}{2}u(t)^\mathsf{T}Ru(t)\,dt \\
\text{s.t.} \quad & \dot{x}(t) = f(x(t),u(t)) := F_x x(t) + F_u u(t) + F_\theta\theta + f_0 \\
& g(x(t),u(t)) := G_x x(t) + G_u u(t) + G_\theta\theta + g_0 \le 0 \\
& h(x(t_\mathrm{f})) := H_x x(t_\mathrm{f}) + H_\theta\theta + h_0 \le 0 \\
& x(t_0) = B_\theta\theta + b_0\,,
\end{aligned} \tag{2.4}$$

where Φ is the optimal value function; $x(t)\in\mathbb{R}^{n_x}$ and $u(t)\in\mathbb{R}^{n_u}$ are the state variables and input variables, respectively, at a given time t; t_0 and t_f are the initial and final times; $g:\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\to\mathbb{R}^{n_g}$ and $h:\mathbb{R}^{n_x}\to\mathbb{R}^{n_h}$ define the path and terminal inequality constraints; and $Q_\mathrm{f}\succeq 0$, $Q\succeq 0$, and $R\succ 0$ are given weighting matrices. We assume that the uncertain parameters $\theta\in\mathbb{R}^{n_\theta}$ appear linearly in the initial conditions, dynamics, path constraints, and terminal constraints of problem (2.4).

The proposed methodology involves two steps, as depicted in Fig. 2.5(b):

- The first (off-line) step defines the multi-parametric control structure, namely mapping the optimal control structure to given measurable quantities, such as the uncertain initial conditions, using mp-DO. This results in a partitioning of the uncertain parameter domain into a number of critical regions, each corresponding to a unique sequence of active path constraints and active terminal constraints. As well as giving a set of conditions for characterizing each critical region, mp-DO determines feedback control laws in the form:

$$u(t) = \mathcal{K}^{(i)}\left(t, \theta, t_1^{(i)}(\theta), \ldots, t_{N_t^{(i)}}^{(i)}(\theta)\right) , \tag{2.5}$$

where the junction times $t_1^{(i)}(\cdot), \ldots, t_{N_t^{(i)}}^{(i)}(\cdot)$ in the optimal switching structure for critical region CR_i are themselves dependent on θ.

- In the subsequent (on-line) step, the multi-parametric NCO-tracking controller is applied in a receding horizon manner. This involves determining the critical regions containing the measured parameters θ and applying the corresponding feedback law until a new measurement becomes available at the next sampling time. Because the switching time functions $t_k^{(i)}(\cdot)$ are typically defined implicitly in practice, even for constrained linear-quadratic control problems, one can either derive fully explicit feedback laws by approximating this functional dependency, or else apply a Newton iteration to compute the $t_k^{(i)}$ for given values of θ at each sampling time.

Both steps are detailed subsequently.

Notation $D_x^\alpha f$ denotes the α-th partial derivative of a function f with respect to x and $f^{(j)}$, the j-th order derivative of with respect to t. The path constraint g_i is said to be of order (or degree) $\sigma_i \geq 0$ if $D_u g_i^{(j)} \equiv 0$ for $j = 0 \ldots \sigma_i - 1$ and $D_u g_i^{(\sigma_i)} \neq 0$, or equivalently,

$$G_{x,i} F_x^{\sigma_i - 1} F_u \neq 0 \quad \text{and} \quad G_{u,i} = G_{x,i} F_x F_u = \cdots = G_{x,i} F_x F^{\sigma_i - 2} F_u = 0 \,.$$

For simplicity, we introduce the notation

$$g_i^{(j)}(x, u) := G_{x,i}^{(j)} x + G_{u,i}^{(j)} x + G_{\theta,i}^{(j)} \theta + g_{0,i}^{(j)} \,,$$

where the row vector $G_{x,i}^{(j)}, G_{u,i}^{(j)}, G_{\theta,i}^{(j)}$ and scalar $g_{0,i}^{(j)}$ can be expressed in terms of F_x, F_u, F_θ, f_θ, G_x, G_u, G_θ and g_0, for each $j = 1 \ldots \sigma_i$. We also make use of the notation

$$g^{(\sigma)}(x, u) := G_x^{(\sigma)} x + G_u^{(\sigma)} u(t) + G_\theta^{(\sigma)} \theta + g_0^{(\sigma)} \,,$$

$$\text{with} \quad G_x^{(\sigma)} := \begin{bmatrix} G_{x,1}^{(\sigma_1)} \\ \vdots \\ G_{x,n_g}^{(\sigma_{n_g})} \end{bmatrix}, \quad G_u^{(\sigma)} := \begin{bmatrix} G_{u,1}^{(\sigma_1)} \\ \vdots \\ G_{u,n_g}^{(\sigma_{n_g})} \end{bmatrix},$$

$$G_\theta^{(\sigma)} := \begin{bmatrix} G_{\theta,1}^{(\sigma_1)} \\ \vdots \\ G_{\theta,n_g}^{(\sigma_{n_g})} \end{bmatrix}, \quad g_0^{(\sigma)} := \begin{bmatrix} g_{0,1}^{(\sigma_1)} \\ \vdots \\ g_{0,n_g}^{(\sigma_{n_g})} \end{bmatrix}.$$

Finally, by a slight abuse of the notation, an over-bar is used to indicate subsets of the terminal or path constraints that are active along a given arc, such as $\bar{g}(x(t), u(t)) = \bar{G}_x x(t) + \bar{G}_u u(t) + \bar{G}_\theta \theta + \bar{g}_0 \leq 0$ and $\bar{\mu}(t)$.

Besides, a list of the notations used for developing multi-parametric solutions in the following sections can be found in Table 2.1.

Table 2.1: List of notations

$x(t)$	continuous-time state variable
$u(t)$	continuous-time control variable
$p(t)$	continuous-time co-state variable
$\mathcal{S}$	switching structure
N_t	the number of arcs
t_k	switching times
$\mu(t)$	multipliers for path constraint
ν	multipliers for terminal constrains
$\mathcal{H}$	Hamiltonian function
π	multipliers at points of discontinuity of $p(t)$
AC_k	sets of active path constraints along the kth arc
NAC_k	sets of inactive path constraints along the kth arc
AC_f	sets of active terminal constraints
NAC_f	sets of inactive terminal constraints
EN_k	sets of path constraints activating at t_k
EX_k	sets of path constraints deactivating at t_k
CO_k	sets of path constraints contacting at t_k

2.3.2 Multi-parametric dynamic optimization

2.3.2.1 Solution structure

For each instance of the parameters θ, the optimal solution of problem (2.4) exhibits a certain switching structure, denoted by $\mathcal{S}(\theta)$. The sequence of active path constraints and active terminal constraints can be characterized by solving the first-order NCO for

Problem (2.4), which come in the form of a multi-point boundary value problem [2]. Under the assumption that the number of arcs $N_t(\theta)$ is finite for each parameter value [78], we denote by $t_k(\theta), k = 1 \dots N_t - 1(\theta)$, the sequence of junction times for each arc in $\mathcal{S}(\theta)$, with $t_0(\theta) = t_0$ and $t_{N_t}(\theta) = t_\mathrm{f}$. These times correspond to the activation or deactivation a particular path constraint or to a touch-and-go point for a higher order path constraint. We denote the sets of path constraints activating, deactivating, or contacting at $t_k(\theta)$ by $\mathrm{EN}_k(\theta)$, $\mathrm{EX}_k(\theta)$ and $\mathrm{CO}_k(\theta)$, respectively. Moreover, $\mathrm{AC}_k(\theta)$ and $\mathrm{NAC}_k(\theta)$ denote the sets of active/inactive path constraints along the kth arc, $t^+_{k-1}(\theta) \leq t \leq t^-_k(\theta)$; and $\mathrm{AC}_\mathrm{f}(\theta)$ and $\mathrm{NAC}_\mathrm{f}(\theta)$, the sets of active/inactive terminal constraints.

Besides its switching structure, characterizing an optimal solution involves determining: (i) the quadruplet of trajectories $(u(t), x(t), p(t), \mu(t))$ along each arc, where $p(t) \in \mathbb{R}^{n_x}$ are the co-state (adjoint) variables, and $\mu(t) \in \mathbb{R}^{n_g}$, the multipliers for the path constraints; (ii) the values of the multipliers $\nu \in \mathbb{R}^{n_h}$ for the terminal constraints; and (iii) the values for the multipliers $\pi^j_{k,i}$ for $j = 1 \dots \sigma_i - 1, i = 1 \dots n_g, k = 1 \dots N_t(\theta) - 1$ at points of discontinuity of the co-state trajectories $p(t)$ (if any). Provided certain controllability and regularity assumptions hold (see below), the following conditions must be satisfied at an optimal solution of Problem (2.4), according to the indirect adjoining approach [61]:

(i) Along each arc $t^+_{k-1}(\theta) \leq t \leq t^-_k(\theta)$, for $k = 1 \dots N_t(\theta)$:

$$\dot{x}(t) = \frac{\partial \mathcal{H}}{\partial p}(u(t), x(t), p(t), \mu(t)) = F_x x(t) + F_u u(t) + F_\theta \theta + f_0 \tag{2.6}$$

$$\dot{p}(t) = -\frac{\partial \mathcal{H}}{\partial x}(u(t), x(t), p(t), \mu(t)) = -Q x(t) - F_x^\mathsf{T} p(t) - {G_x^{(\sigma)}}^\mathsf{T} \mu(t) \tag{2.7}$$

$$0 = \frac{\partial \mathcal{H}}{\partial u}(u(t), x(t), p(t), \mu(t)) = R u(t) + F_u^\mathsf{T} p(t) + {G_u^{(\sigma)}}^\mathsf{T} \mu(t) \tag{2.8}$$

$$0 = \mu_i(t) g_i(x(t), u(t)) = \mu_i(t)\ \left(G_{x,i} x(t) + G_{u,i} u(t) + G_{\theta,i}\theta + g_{0,i}\right) \tag{2.9}$$

$$(-1)^j \mu_i^{(j)}(t) \geq 0 \geq g_i(x(t), u(t)) = G_{x,i} x(t) + G_{u,i} u(t) + G_{\theta,i}\theta + g_{0,i}\,, \tag{2.10}$$

for each $i = 1 \dots n_g$ and each $j = 1 \dots \sigma_i$, and with the Hamiltonian function

$$\begin{aligned} \mathcal{H}(u, x, p, \mu) := {} & \frac{1}{2} x^\mathsf{T} Q x + \frac{1}{2} u^\mathsf{T} R u + p^\mathsf{T}(F_x x + F_u u + F_\theta \theta + f_0) \\ & + \mu^\mathsf{T} \left(G_x^{(\sigma)} x + G_u^{(\sigma)} u + G_\theta^{(\sigma)} \theta + g_0^{(\sigma)}\right) . \end{aligned} \tag{2.11}$$

(ii) At the terminal time $t_\mathrm{f} = t_{N_t(\theta)}(\theta)$:

$$p(t_\mathrm{f}) = Q_\mathrm{f} x(t_\mathrm{f}) + H_x^\mathsf{T} \nu \tag{2.12}$$

$$0 = \nu_i h_i(x(t_\mathrm{f})) = \nu_i \left(H_{x,i} x(t_\mathrm{f}) + H_{\theta,i}\theta + h_{0,i} \right) \tag{2.13}$$

$$\nu_i \geq 0 \geq h_i(x(t_\mathrm{f})) = H_{x,i} x(t_\mathrm{f}) + H_{\theta,i}\theta + h_{0,i}\,, \tag{2.14}$$

for each $i = 1 \ldots n_h$.

(iii) At each junction time $t_k(\theta)$, for $k = 1 \ldots N_t(\theta) - 1$:

$$\mathcal{H}(u(t_k^-), x(t_k), p(t_k^-), \mu(t_k^-)) = \mathcal{H}(u(t_k^+), x(t_k), p(t_k^+), \mu(t_k^+)) \tag{2.15}$$

$$p(t_k^-) = p(t_k^+) + \sum_{i=1}^{n_g} \sum_{j=1}^{\sigma_i} \pi_{k,i}^j D_x g_i^{(j)}(x(t_k), u(t_k^+)) = p(t_k^+) + \sum_{i=1}^{n_g} \sum_{j=1}^{\sigma_i} \pi_{k,i}^j G_{x,i}^{(j)} \tag{2.16}$$

$$0 = \pi_{k,i}^j g_i(x(t_k), u(t_k^+)) \tag{2.17}$$

$$\pi_{k,i}^j \begin{cases} \geq (-1)^{\sigma_i - 1} \mu_i^{(\sigma_i - 1)}(t_k^+), & \text{if } i \in \mathrm{EN}_k(\theta) \cup \mathrm{CO}_k(\theta) \text{ and } j = 1 \\ = (-1)^{\sigma_i - j} \mu_i^{(\sigma_i - j)}(t_k^+), & \text{if } i \in \mathrm{EN}_k(\theta) \text{ and } j > 1 \\ = 0 & \text{otherwise} \end{cases} \tag{2.18}$$

$$\mu_i^{(\sigma_i - j)}(t_k^-) = 0, \quad \text{if } i \in \mathrm{EX}_k(\theta) \cup \mathrm{CO}_k(\theta) \text{ and } 0 \leq j \leq \sigma_i - 2, \tag{2.19}$$

for each $i = 1 \ldots n_g$ and each $j = 1 \ldots \sigma_i$.

Note that the multipliers π may only appear in the optimal conditions for problems with pure state path constraints of order 1 or higher; they can be discarded in problems having mixed control-state path constraints only, where the adjoint trajectories are continuous at junction times.

In general, the foregoing optimality conditions (2.6)-(2.19) are only necessary under the additional assumptions that: (i) the pair (F_x, F_u) is controllable, which precludes abnormality [79]; and (ii) both the active path and active terminal constraints are regular [61],

$$\operatorname{rank} \begin{bmatrix} \bar{G}_u^{(\sigma)} & g(x(t), u(t)) \end{bmatrix} = n_g, \quad t_{k-1}^+(\theta) \leq t \leq t_k^-(\theta), \quad k = 1 \ldots N_t(\theta), \quad \text{and}$$

$$\operatorname{rank} \begin{bmatrix} H_x & h(x(t_\mathrm{f})) \end{bmatrix} = n_h\,.$$

Moreover, under the extra assumption of strict complementarity slackness for the multipliers ν, $\pi^j_{k,i}$ and $\mu(t)$ along each arc $t^+_{k-1}(\theta_0) \le t \le t^-_k(\theta_0)$ for a given parameter value θ_0, and by strict convexity of the objective function and linearity of the dynamics and constraints, the optimal trajectories $u(t)$, $x(t)$, $p(t)$, $\mu(t)$ for $t^+_{k-1}(\theta_0) \le t \le t^-_k(\theta_0)$, optimal multipliers ν and $\pi^j_{k,i}$, and optimal switching/contact times t_k describe differentiable functions in an (open) neighborhood of θ_0 [57–59]; see also [49]. Expressions for these functions are established in the following subsection.

2.3.2.2 Feedback control laws

Using the previous optimality conditions, explicit feedback control laws can be derived along each arc of the optimal solution. Using condition (2.8), together with the fact that $\bar{g}^{(\sigma)}(x(t), u(t)) = \bar{G}^{(\sigma)}_x x(t) + \bar{G}^{(\sigma)}_u u(t) + \bar{G}^{(\sigma)}_\theta \theta + \bar{g}^{(\sigma)}_0 = 0$ along an arc, we have

$$\bar{\mu}(t) = \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \left[\bar{G}^{(\sigma)}_x x(t) - \bar{G}^{(\sigma)}_u R^{-1} F^\mathsf{T}_u p(t) + \bar{G}^{(\sigma)}_\theta \theta + \bar{g}^{(\sigma)}_0\right] \tag{2.20}$$

$$u(t) = -R^{-1}\left[F^\mathsf{T}_u p(t) + \bar{G}^{(\sigma)\mathsf{T}}_u\, \bar{\mu}(t)\right] \tag{2.21}$$

which are both well-defined under the assumption that $\bar{G}^{(\sigma)}_u$ is full rank. In turn, the state and co-state equations (2.6)-(2.7) can be rewritten in the form

$$\begin{bmatrix} \dot{x}(t) \\ \dot{p}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \Phi^{(k)}_{xx} & \Phi^{(k)}_{xp} \\ \Phi^{(k)}_{px} & \Phi^{(k)}_{pp} \end{bmatrix}}_{A^{(k)}_{xp}} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \Phi^{(k)}_{x\theta} \\ \Phi^{(k)}_{p\theta} \end{bmatrix}}_{A^{(k)}_\theta} \theta + \underbrace{\begin{bmatrix} \varphi^{(k)}_{x0} \\ \varphi^{(k)}_{p0} \end{bmatrix}}_{a^{(k)}_0} \tag{2.22}$$

$$\begin{aligned}
\text{with:}\quad \Phi^{(k)}_{xx} &:= F_x - F_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{G}^{(\sigma)}_x \\
\Phi^{(k)}_{xp} &:= -\left[F_u - F_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{G}^{(\sigma)}_u\right] R^{-1} F^\mathsf{T}_u \\
\Phi^{(k)}_{x\theta} &:= F_\theta - F_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{G}^{(\sigma)}_\theta \\
\varphi^{(k)}_{x0} &:= f_0 - F_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{g}^{(\sigma)}_0 \\
\Phi^{(k)}_{px} &:= -Q - \bar{G}^{(\sigma)\mathsf{T}}_x \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{G}^{(\sigma)}_x \\
\Phi^{(k)}_{pp} &:= -\Phi^{(k)}_{xx} \\
\Phi^{(k)}_{p\theta} &:= -\bar{G}^{(\sigma)\mathsf{T}}_x \left[\bar{G}^{(\sigma)}_u R^{-1}\, \bar{G}^{(\sigma)\mathsf{T}}_u\right]^{-1} \bar{G}^{(\sigma)}_\theta
\end{aligned}$$

$$\varphi_{p0}^{(k)} := \; - \, \bar{G}_x^{(\sigma)\,\mathsf{T}} \left[\bar{G}_u^{(\sigma)} R^{-1} \, \bar{G}_u^{(\sigma)\,\mathsf{T}} \right]^{-1} \bar{g}_0^{(\sigma)} \, .$$

This way, we may express $x(t)$ and $p(t)$ at each $t \in [t_{k-1}^{+}(\theta), t_k(\theta)^{-}]$, and therefore also $u(t)$ and $\mu(t)$, as parametric functions of the uncertainty θ, the junction times t_k, and the state/co-state values at t_k:

$$\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = \exp\left(A_{xp}^{(k)} [t - t_{k-1}(\theta)] \right) \begin{bmatrix} x(t_{k-1}(\theta)) \\ p(t_{k-1}^{+}(\theta)) \end{bmatrix} + \int_{t_{k-1}(\theta)}^{t} \exp\left(A_{xp}^{(k)} [t - \tau] \right) \left[A_\theta^{(k)} \theta + a_\theta^{(k)} \right] \, d\tau \, . \tag{2.23}$$

In the case that $A_{xp}^{(k)}$ is nonsingular, we have:

$$\begin{aligned} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = & \exp\left(A_{xp}^{(k)} [t - t_{k-1}(\theta)] \right) \left(\begin{bmatrix} x(t_{k-1}(\theta)) \\ p(t_{k-1}^{+}(\theta)) \end{bmatrix} + \left(A_{xp}^{(k)} \right)^{-1} \left[A_\theta^{(k)} \theta + a_\theta^{(k)} \right] \right) \\ & - \left(A_{xp}^{(k)} \right)^{-1} \left[A_\theta^{(k)} \theta + a_\theta^{(k)} \right] \, . \end{aligned} \tag{2.24}$$

When $A_{xp}^{(k)}$ is singular, an explicit expression can be obtained by considering the normal Jordan form of $A_{xp}^{(k)}$ instead.

At this point, parametric expressions for the terminal and interior-point constraint multipliers ν and π_k can be obtained by piecing together (2.23) on $[t_0, t_\mathrm{f}]$ and exploiting the equality conditions in (2.12)-(2.13) and (2.16)-(2.19). In the case of mixed state-input path constraints only, the optimal state and co-state trajectories are both continuous at the junction times. Then, since $\bar{H}_x$ is full rank, and provided that $A_{xp}^{(k)}$ is invertible on each arc, parametric expressions for the active terminal constraint multipliers $\bar{\nu}$, terminal state $x(t_\mathrm{f})$ and initial adjoint $p(t_0)$ can be obtained from the following linear system:

$$0 = \bar{H}_x x(t_\mathrm{f}) + \bar{H}_\theta \theta + \bar{h}_0 \tag{2.25}$$

$$\begin{aligned} \begin{bmatrix} x(t_\mathrm{f}) \\ Q_\mathrm{f} x(t_\mathrm{f}) + \bar{H}_x^\mathsf{T} \bar{\nu} \end{bmatrix} = & \exp\left(\sum_{k=1}^{N_t(\theta)} A_{xp}^{(k)} [t_k(\theta) - t_{k-1}(\theta)] \right) \begin{bmatrix} B_\theta \theta + b_0 \\ p(t_0) \end{bmatrix} \\ & + \sum_{k=1}^{N_t(\theta)} \exp\left(\sum_{j=k+1}^{N_t(\theta)} A_{xp}^{(j)} [t_j(\theta) - t_{j-1}(\theta)] \right) \\ & \left[\exp\left(A_{xp}^{(k)} [t_k(\theta) - t_{k-1}(\theta)] \right) - I \right] \left(A_{xp}^{(k)} \right)^{-1} \left[A_\theta^{(k)} \theta + a_0^{(k)} \right] . \end{aligned} \tag{2.26}$$

(In the case of a single control arc, $N_t(\theta) = 1$, the term $\sum_{j=k+1}^{N_t(\theta)} A_{xp}^{(j)} [t_j(\theta) - t_{j-1}(\theta)]$ in the right-hand side of (2.26) cancels to zero.)

Overall, for a given structure $\mathcal{S}(\theta)$, the solution of Problem (2.4) can therefore be expressed in parametric form as

$$(u(t), x(t), p(t), \mu(t), \nu, \pi) = \mathcal{F}_{\mathcal{S}(\theta)}\left(t, \theta, t_1(\theta), \ldots, t_{N_t(\theta)-1}(\theta)\right) . \tag{2.27}$$

Naturally, this construction can be automated in a practical implementation. One could also exploit the remaining optimality conditions (2.15) in order to determine parametric expressions of the junction times t_k as a function of θ alone. Explicit expressions are often not possible, however, due to the inherent nonlinearity of the parametric state/co-state expressions (2.23) in t_k and θ. In practice, one may either use approximate explicit expressions for $t_k(\theta)$, or compute these junction times on-line using a Newton iteration. These considerations are discussed further in Section 2.3.3.

2.3.2.3 Critical regions

Each critical region corresponds to a subset of the uncertain parameter domain $\Theta \subseteq \mathbb{R}^{n_\theta}$, whereby the corresponding optimal control solutions all share the same switching structure in terms of the sequence of active path constraints and active terminal constraints. Formally, given the switching structure $\mathcal{S}$ comprised of N_t arcs with corresponding index sets EN_k, EX_k, CO_k, AC_k, NAC_k, $\mathrm{AC_f}$, and $\mathrm{NAC_f}$, the critical region $\mathrm{CR}_\mathcal{S}$ associated with $\mathcal{S}$ is defined as

$$\mathrm{CR}_\mathcal{S} := \left\{ \theta \in \Theta \;\middle|\; \begin{array}{l} \exists x(\cdot), u(\cdot), p(\cdot), \mu(\cdot), \nu, \pi, t_1, \ldots, t_{N_t-1} : \\ (u(t), x(t), p(t), \mu(t), \nu, \pi) = \mathcal{F}_\mathcal{S}\left(t, \theta, t_1, \ldots, t_{N_t-1}\right) \\ t_0 \leq t_1 \leq \cdots \leq t_{N_t-1} \leq t_\mathrm{f} \\ \mathcal{H}(u(t_k^-), x(t_k), p(t_k^-), \mu(t_k^-)) = \mathcal{H}(u(t_k^+), x(t_k), p(t_k^+), \mu(t_k^+))\,, \;\; k = 1 \ldots N_t - 1 \\ (-1)^j \mu_i^{(j)}(t) \geq 0, \quad i \in \mathrm{AC}_k,\; t \in [t_{k-1}^+, t_k^-],\; k = 1 \ldots N_t \\ g_j(x(t), u(t)) \leq 0, \quad j \in \mathrm{NAC}_k,\; t \in [t_{k-1}^+, t_k^-],\; k = 1 \ldots N_t \\ \pi_{k,i}^1 \geq (-1)^{\sigma_i - 1} \mu_i^{(\sigma_i - 1)}(t_k^+) \;\text{ if } \sigma_i > 0, \quad i \in \mathrm{EN}_k \cup \mathrm{CO}_k,\; k = 1 \ldots N_t - 1 \\ \nu_i \geq 0, \quad i \in \mathrm{AC_f} \\ h_j(x(t_\mathrm{f})) \leq 0, \quad j \in \mathrm{NAC_f} \end{array} \right\} \tag{2.28}$$

The boundary between two critical regions thus corresponds to either an inactive terminal or path constraint activating, or an active terminal or path constraint inactivating, or two subsequent junction times becoming equal.

Similar to the procedure outlined in Section 2.2.1 for the construction of critical regions in mp-MPC, a systematic procedure for constructing critical regions in mp-DO is as follows:

0. Define the uncertainty domain $\Theta \subset \mathbb{R}^{n_\theta}$, and set $N = 0$.
1. Select a point $\theta \in \Theta \setminus \cup_{i=1}^{N} \mathrm{CR}^{(i)}$ that is feasible for (2.4). If no such point exists, stop; else, set $N \leftarrow N + 1$.
2. Compute a solution to the control problem (2.4), and characterize the corresponding switching structure $\mathcal{S}^{(N)} := \mathcal{S}(\theta)$.
3. Define the new critical region $\mathrm{CR}^{(N)} := \mathrm{CR}_{\mathcal{S}(\theta)}$, with corresponding feedback laws $\mathcal{F}^{(N)} := \mathcal{F}_{\mathcal{S}(\theta)}$.
4. Return to step 1.

On termination, this procedure returns the number N of critical regions contained in the initial domain Θ, together with the solution structure $\mathcal{S}^{(i)}$ and the feedback laws $\mathcal{F}^{(i)}$ in each region $\mathrm{CR}^{(i)}$, $i = 1 \ldots N$, as defined implicitly by (2.28). A number of remarks are in order regarding the practical implementation of this procedure:

- The selection of a point $\theta \in \Theta \setminus \cup_{i=1}^{N} \mathrm{CR}^{(i)}$ in step 1 is a difficult problem in general, since (2.28) may describe non-convex subsets. Suppose, without loss of generality, that the critical regions can be written in the form $\mathrm{CR}^{(i)} = \{\theta \in \Theta \,|\, \Gamma_j^{(i)}(\theta) \leq 0, j = 1 \ldots M^{(i)}\}$, for given inequality constraint functions $\Gamma_1^{(i)}, \ldots, \Gamma_{M^{(i)}}^{(i)}$. Then, finding a point $\theta \in \Theta \setminus \cup_{i=1}^{N} \mathrm{CR}^{(i)}$ amounts to finding a feasible point for the disjunctive constraint

$$\bigwedge_{i=1}^{N} \left(\bigvee_{j=1}^{M^{(i)}} \Gamma_j^{(i)}(\theta) > 0 \right) \wedge (\theta \in \Theta) = \text{true}\,. \tag{2.29}$$

  In particular, techniques based on complete search have become available in recent years to address such constraint satisfaction problems; see, e.g., [80]. More and more constraints are appended as the number of critical regions found increases, and the problem eventually becomes infeasible when the parameter region has been exhausted.

- Step 2 involves solving the constrained dynamic optimization problem (2.4) and characterizing its underlying solution structure. A numerical solution can be computed using direct solution methods [81], which discretize the control and/or state variables in order to arrive at an approximate, finite-dimensional optimization problem. In the present case, this approximate problem is a convex QP for which efficient and reliable large-scale solvers are available. Moreover, such direct solution methods can be used in combination with adaptive structure detection techniques, as discussed e.g. in [39, 82].

The computation of critical regions and characterization of the corresponding feedback laws is now illustrated for the linear mp-MPC problem first considered in Section 2.2.1.

Illustrative example (continued) We consider a mp-DO problem for the second-order dynamic system (2.2) in the form of Problem (2.4), with

$$F_x = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, \quad F_u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad G_u = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad g_0 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \quad B_\theta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and F_θ, f_0, G_x, G_θ, H_x, H_θ and h_0 all zero. This mp-DO aims to solve the same optimization-based control problem as in (2.3), yet in continuous-time form—that is, without discretizing the time horizon and the dynamics. The solution for $\Theta = [-2, 2]^2$ using the approach described in this subsection yields $N = 3$ critical regions, as shown in Fig. 2.6:

- The optimal control strategy for an initial state in the critical region CR01 is comprised of a unique unconstrained arc, which is therefore identical to classical unconstrained LQR;

- For an initial state in CR02, the optimal solution structure is comprised of two arcs, namely a constrained arc with the input at its lower bound followed by an unconstrained arc;

- For an initial state in CR03 likewise, the optimal solution consists of a constrained arc with the input at its upper bound followed by an unconstrained arc.

Expressions for the feedback laws $\mathcal{F}^{(i)}$ in each region $\mathrm{CR}^{(i)}$, $i = 1\ldots 3$ are derived as follows:

- We start with the point $\theta = (0,0)$, whose corresponding optimal control solution is unconstrained. We denote by $\mathcal{S}^{(1)}$ this solution structure, which is comprised of a single arc ($N_t^{(1)} = 1$). Expressions for the state and adjoint trajectories as a function of θ are obtained directly from (2.24), with the missing initial adjoints determined from (2.26) as

$$\begin{bmatrix} x(t_0) \\ p(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1.6396 & 1.6396 \\ 1.6396 & 21.6396 \end{bmatrix} \theta\,.$$

  An expression of $u(t)$ follows from (2.21) as

$$u(t,\theta) = \begin{bmatrix} -8.198 & -8.198 \end{bmatrix} \theta \, \exp(-10.198t), \tag{2.30}$$

  Finally, an implicit characterization of the critical region $\mathrm{CR}^{(1)}$ corresponding to $\mathcal{S}^{(1)}$ is obtained from (2.28) which, in the absence of active constraints, corresponds to values of θ for which both input constraints remain inactive:

$$\mathrm{CR}^{(1)} \;:=\; \left\{ \theta \in [-2,2]^2 \;\middle|\; \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \theta \leq \begin{bmatrix} 0.2440 \\ 0.2440 \end{bmatrix} \right\}.$$

- The point $\theta = (1,1)$ happens to be in the uncertainty domain Θ, yet outside $\mathrm{CR}^{(1)}$. The structure $\mathcal{S}^{(2)}$ of the optimal control at this point is comprised of two arcs ($N_t^{(2)} = 2$), namely a boundary arc corresponding to an active lower input bound, followed by an interior arc. The determination of an explicit feedback control law starts by computing the adjoint initial condition $p(t_0)$ as a function of θ from (2.26), then expressing the state/co-state trajectories as a function of θ by (2.24), and finally using (2.21). In this specific case, an explicit expression of the switching time function $t_1(\theta)$ can be obtained by expressing continuity of the optimal control at t_1 as

$$t_1(\theta) = 0.5\ln(0.8039(1+\theta_1+\theta_2))\,,$$

and an expression of the feedback control $u(t)$ is given by

$$u(t,\theta) = \begin{cases} -2, & \text{if } t \le t_1(\theta) \\ -2\exp(-18.396[t-t_1(\theta)]), & \text{otherwise} \end{cases}$$

An implicit characterization of this critical region is obtained from (2.28) as

$$\mathrm{CR}^{(2)} := \left\{ \theta \in [-2,2]^2 \;\middle|\; \begin{bmatrix} -1 & -1 \end{bmatrix} \theta \le -0.2440 \right\}.$$

- The point $\theta = (-1,-1)$ happens to be in the uncertainty domain Θ, yet outside $\mathrm{CR}^{(1)} \cup \mathrm{CR}^{(2)}$. The structure $\mathcal{S}^{(3)}$ of the optimal control at this point is also comprised of two arcs ($N_t^{(3)} = 2$), namely a boundary arc corresponding to an active upper input bound, followed by an interior arc. The determination of an explicit feedback control law and the characterization of $\mathrm{CR}^{(3)}$ are similar to the previous case:

$$u(t,\theta) = \begin{cases} 2, & \text{if } t \le t_1(\theta) \\ 2\exp(-18.396[t-t_1(\theta)]), & \text{otherwise} \end{cases}$$

with $t_1(\theta) = 0.5\ln(0.8039(1-\theta_1-\theta_2))$, and

$$\mathrm{CR}^{(3)} := \left\{ \theta \in [-2,2]^2 \;\middle|\; \begin{bmatrix} 1 & 1 \end{bmatrix} \theta \le -0.2440 \right\}.$$

- At this point, we have $\mathrm{CR}^{(1)} \cup \mathrm{CR}^{(2)} \cup \mathrm{CR}^{(3)} = \Theta$ and the procedure terminates with a total of three critical regions.

For instance, the open-loop optimal trajectories for the initial states $\theta^{(1)} = (1,-0.8) \in$ CR01 and $\theta^{(2)} = (1,1) \in$ CR02 are shown in Fig. 2.7(a) and 2.7(b), respectively. Note that an explicit expression for the switching time t_1 in both CR02 and CR03 as a function of the initial conditions θ can also be found for this simple problem as

$$t_1(\theta) := \begin{cases} 0.5\ln(0.8039(1+\theta_1+\theta_2)), & \theta \in \mathsf{CR02} \\ 0.5\ln(0.8039(1-\theta_1-\theta_2)), & \theta \in \mathsf{CR03} \end{cases}$$

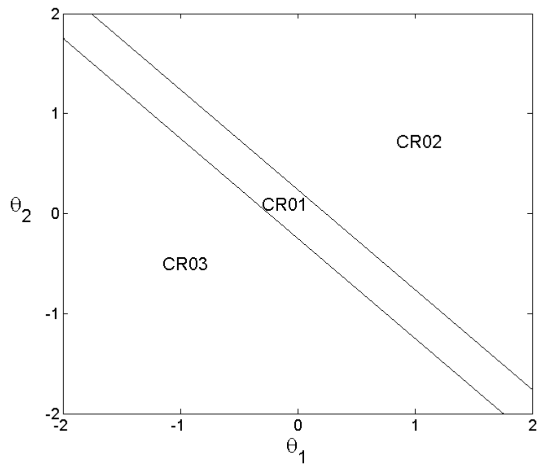
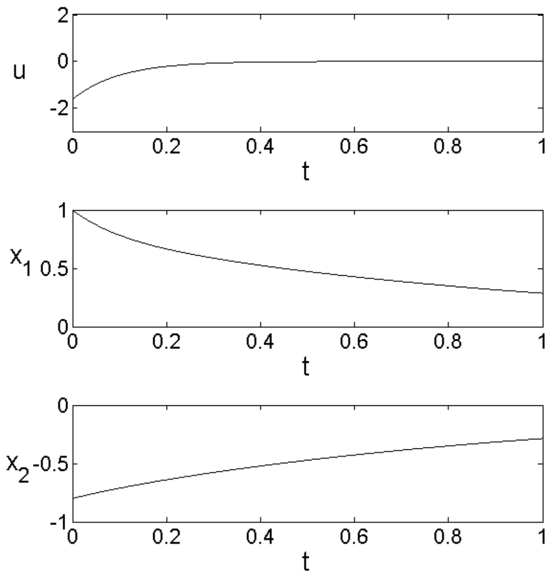


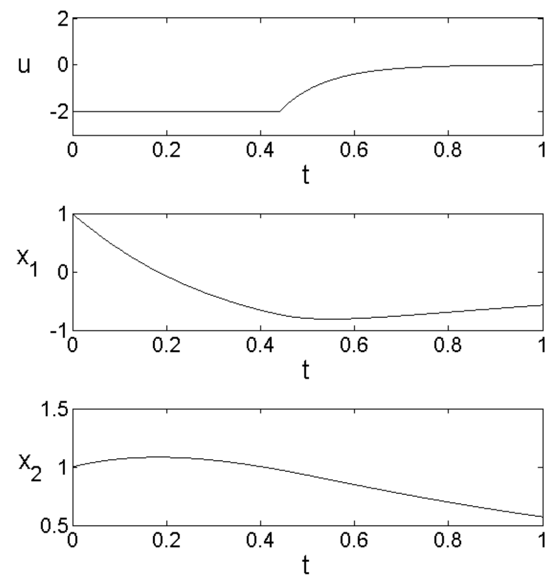
Figure 2.6: Critical regions

The critical regions of the mp-DO problem are closely related to those of the discretized mp-MPC problem in Fig. 2.2. First of all, the central region CR01 in Fig. 2.6 matches the central regions CR08 and CR15 in Fig. 2.2(a) and 2.2(b), respectively, and all three feedback laws correspond to the same unconstrained LQR strategy, $u(t) = [8.198 \;\; 8.198]x(t)$. The rest of the critical regions in Fig. 2.2 have their first few control stages either at the upper bound or at the lower bound, and the farther the central region, the larger the number of time intervals with active constraints. This behavior is fully consistent with the switching time t_1 between the upper/lower bound and the interior arc in the continuous-time optimal control increasing in moving further away from CR01. By construction, the feedback laws corresponding to the critical regions other than the central region in Fig. 2.2 are approximations of the feedback laws computed for the critical regions CR02 and CR03 in Fig. 2.6. The finer the discretization, the closer the approximate feedback laws to the continuous-time ones; but this comes at the price of a significant increase in the number of critical regions in order to better capture the variations in switching time, about double the number from $N = 10$ to $N = 20$.

In this interpretation, mp-DO thus provides a means of reducing the number of critical regions by accounting for the underlying switching structure and introducing the switching times in the feedback law parameterization. This reduction becomes more effective as



(a) $\theta^{(1)} = (1, -0.8) \in \mathsf{CR01}$



(b) $\theta^{(2)} = (1, 1) \in \mathsf{CR02}$

Figure 2.7: Optimal solution of mp-DO in illustrative example.

certain arcs in the mp-DO solution spans many time intervals in the discretized mp-MPC problem.

2.3.3 Multi-parametric NCO-tracking controller

The critical regions $\mathrm{CR}^{(1)}, \ldots, \mathrm{CR}^{(N)}$ and their corresponding feedback laws $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(N)}$ define a multi-parametric NCO-tracking controller for the problem (2.4) at hand. At this point, we like to reiterate that the feedback laws are determined off-line via a (continuous-time) mp-DO formulation, whereas the closed-loop mp-NCO-tracking controller selects the appropriate control law after the uncertainty has been revealed and applies them in a receding horizon manner. In this respect, mp-QP and mp-MPC are the discrete-time counterparts of mp-DO and mp-NCO-tracking, respectively.

The application of the mp-NCO-tracking controller, in a receding horizon manner, involves determining the critical region corresponding to the uncertainty revealed at the current sampling time. Checking whether or not $\theta \in \mathrm{CR}^{(i)}$ for a given $i = 1 \ldots N$ can be done in two steps:

1. Given the feedback laws $\mathcal{F}^{(i)}$, determine the switching times $t_1(\theta), \ldots, t_{N_t^{(i)}(\theta)-1}(\theta)$ satisfying the Hamiltonian continuity conditions (2.15);
2. Test whether all the auxiliary inequalities defining $\mathrm{CR}^{(i)}$ as in (2.28) are satisfied.

Once the correct critical region CR^* has been identified, the controller simply applies the associated feedback law $\mathcal{F}^*$ until the following sampling time.

In this approach, the switching times may be computed using a Newton iteration (or a robustified variant such as the dog-leg method). The on-line computational burden can be reduced by initializing the switching times with the values at the previous sampling time. This warm-starting strategy is most effective when the sampling frequency is fast, so the variations in switching times between two sampling times remain small. Moreover, since only the first part of the optimal feedback law is applied in a receding horizon strategy, the same control law may be applied several times when not foreseeing any switching time nearby. Detection of such instances can lead to large computational-time reductions. An alternative approach involves approximating the functional dependency of the switching times with respect to the uncertainty, for instance using linear or polynomial functions in θ. These extra laws can be determined via parametric programming too, possibly after further partitioning of the critical regions for keeping the approximation level under control.

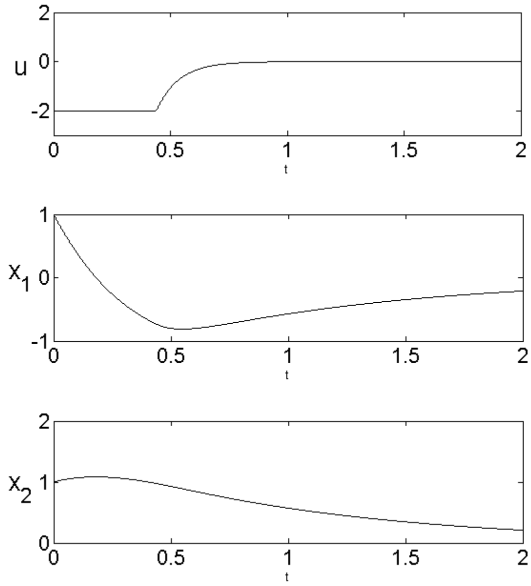
The application of a multi-parametric NCO-tracking controller for the same linear mp-MPC problem as in Section 2.2.1 is presented below.

Illustrative example (continued) We follow the steps presented in Section 2.3.2.3 in order to determine the critical regions in the uncertainty domain $\Theta := [-2, 2]^2$.

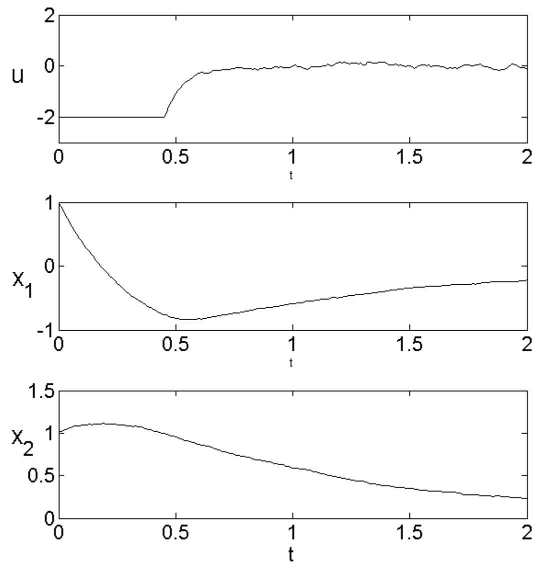
The critical regions shown in Fig. 2.6 define a multi-parametric NCO-tracking controller for the linear system (2.2). The application of this controller in a receding horizon scheme is illustrated in Fig. 2.8, for a sampling period of $T = 10^{-2}$ and from the initial state $\theta^{(2)} = (1, 1) \in \mathsf{CR02}$. The control and state trajectories in the noise-free case (Fig. 2.8(a)) closely match those of the nominal, open-loop case in Fig. 2.7(b). A difference here is the value of the switching time t_1, which is postponed with the closed-loop controller due to the finite sampling period. The closed-loop trajectories for the same problem, now with Gaussian white noise with signal-to-noise ratio of 50dB added to the state measurements at each sample time, are shown in Fig. 2.8(b) for comparison. Recall also that explicit expressions of the switching time t_1 as a function of θ can be obtained in both critical regions $\mathsf{CR02}$ and $\mathsf{CR03}$, so no Newton iteration or approximation is needed in this instance.

2.3.4 Computational aspects

The computational burden for the proposed multi-parametric NCO-tracking framework involves two distinct components, namely the off-line controller design and its on-line application. The former is dominated by the solution of an mp-DO problem, where all the variables are linear functions of the parameters, but the switching times. Because of this nonlinearity, the critical regions do not describe convex polytopes anymore, but yield general non-convex regions. Even though the explicit characterization of a region's boundary is not needed, the enumeration procedure in Section 2.3.2.3 calls for the application of complete search methods in order to find a new critical region or to establish that all the critical regions have been mapped already, which can be computationally demanding (if at all tractable) [80]. In practice, one can apply model reduction techniques for reducing the order of the dynamic model and improve the computational tractability



(a) Noise-free case



(b) With Gaussian white noise

Figure 2.8: Control and response of the multi-parametric NCO-tracking controller in illustrative example.

of the mp-DO problem, yet without causing too big a performance loss for the resulting controllers [83, 84].

With regards to the on-line application aspects, mp-NCO-tracking controllers provably reduce the number of critical region compared with their discretized mp-QP counterparts. Nonetheless, the on-line computational burden with mp-NCO-tracking is typically dominated by the determination of the switching times corresponding to the measured uncertainty θ, e.g. using a Newton iteration. In practice, efficient warm-starting strategies could be developed for minimizing this burden, especially for short sampling periods. Whether or not such strategies will make mp-NCO-tracking competitive with simple look-up table evaluation in discretized mp-QP despite a possibly much larger number of critical regions will be explored as part of future work.

2.4 Numerical case studies

The objective of the numerical case studies in this section is two-fold: (i) illustrate the computation of mp-DO critical regions for problems with first- and higher-order state path constraints along with terminal constraints, thereby complementing the foregoing illustrative example with simple bound constraints only (Sects. 2.4.1 and 2.4.2); and (ii) present the mp-DO construction and the corresponding multi-parametric NCO-tracking controller for a more challenging FCC unit with multiple inputs (Section 2.4.3). The computational effort for solving the mp-DO problem mainly depends on the complexity of the multi-parametric switching structures, which corresponds to the critical regions. For the following case studies, the numbers of the critical regions are no more than 11 and all the computations for the multi-parametric solutions are completed within a few seconds.

2.4.1 Critical regions in mp-DO problem with first-order state path constraints

We consider the following problem:

$$
\begin{aligned}
\min_{\substack{u(t),x_1(t),x_2(t),\\ 0\le t\le 1}} \ & \frac{1}{2}\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0.1242 & 0.0846 \\ 0.0846 & 0.9854 \end{bmatrix} \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} \\
& + \int_0^1 \frac{1}{2}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 1 & 0 \\ 0 & 0.8 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \frac{0.01}{2} u(t)^2 \, dt \\
\text{s.t.} \ & \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 1.5 & -0.5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\
& \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\
& 1.5x_1(t) + x_2(t) \le 2 \,,
\end{aligned}
\tag{2.31}
$$

where the path constraint $g(x) := 1.5x_1 + x_2 - 2 \le 0$ is of first order with respect to dynamics. Moreover, the uncertainty is in the initial state conditions only, and the corresponding uncertainty domain is chosen as $\Theta := [-8, 8] \times [-10, 0]$.

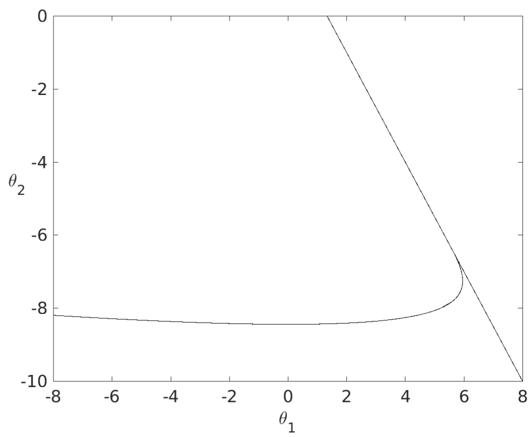


Figure 2.9: Critical regions of the mp-DO problem (2.31).

Two critical regions found on application of the algorithm in Section 2.2.1 are shown in Fig. 2.9, where part of the uncertainty domain is discarded due to infeasibility. The structure of the optimal control solutions in the critical region CR_1 consist of a unique interior arc, whereas those solutions in CR_2 are comprised of three arcs, an interior arc, followed by a boundary arc where the path constraint is binding, and a final interior arc

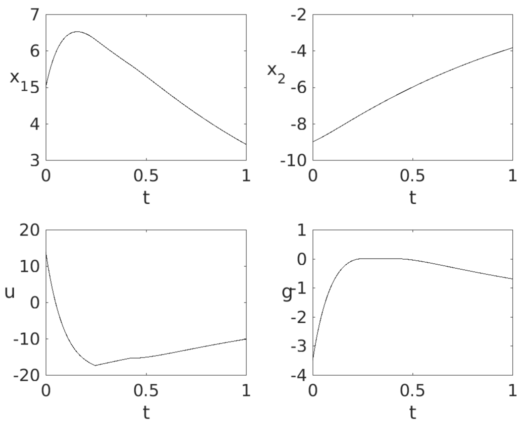


Figure 2.10: Optimal solution of the mp-DO problem (2.31) for $\theta = (5, -9) \in \mathsf{CR}_2$.

constrained arc sharing the same control law as in CR_1. Here, the boundary between CR_1 and CR_2 consists of those optimal control solutions where the path constraint activates at a single critical point along the time horizon $[0, 1]$, and turns out to be nonlinear. For illustration, the optimal control and response trajectories along with the path constraint profile are shown in Fig. 2.10 for the uncertainty realization $\theta = (5, -9)$ in the critical region CR_2; the path constraint is active over the time interval $[0.2465, 0.4205]$.

2.4.2 Critical regions in mp-DO problem with second order state constraints

We now consider the following problem:

$$
\begin{aligned}
&\min_{\substack{u(t),x_1(t),x_2(t),\\ 0\le t\le 1}} \int_0^1 \frac{1}{2} u(t)^2\, dt\\
&\text{s.t.} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)\\
&\qquad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ \theta_2 \end{bmatrix}\\
&\qquad -\theta_1 \le x_1(t) \le \theta_1\\
&\qquad x_1(1) \le 0, \quad x_2(1) \le \theta_2\,,
\end{aligned}
\tag{2.32}
$$

where the path constraints are of second order. Here, uncertainty $\theta \in \Theta := [0, 1] \times [-1, 1]$ is present both in the initial conditions and the path/terminal constraints.

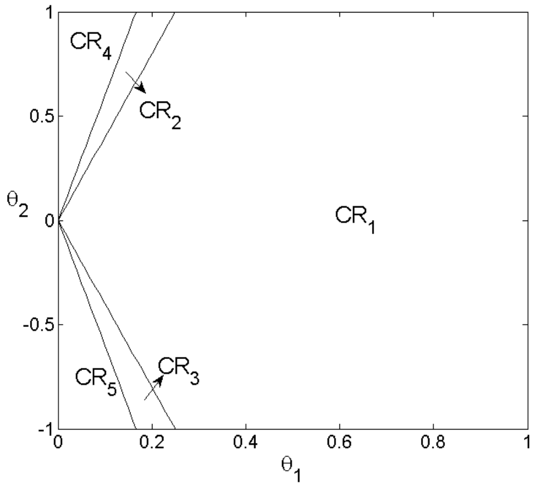
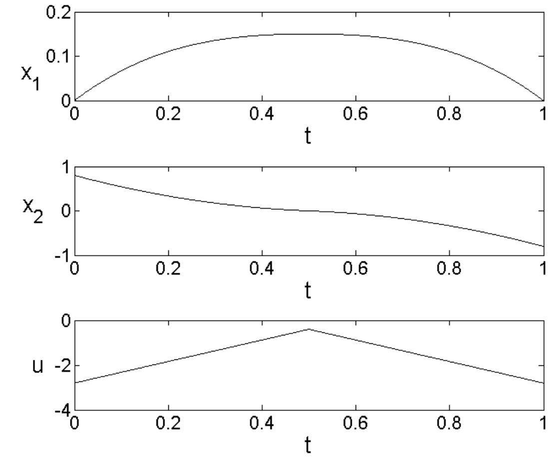
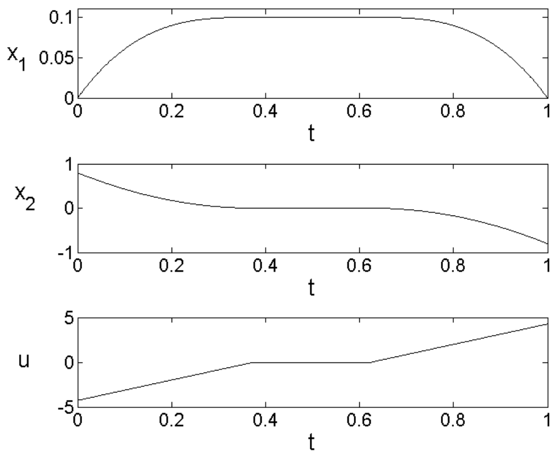


Figure 2.11: Critical regions of the mp-DO problem (2.32).



(a) $\theta^{(1)} = (0.15, 0.8) \in \mathsf{CR}_2$



(b) $\theta^{(2)} = (0.1, 0.8) \in \mathsf{CR}_4$

Figure 2.12: Optimal solution of the mp-DO problem (2.32).

A total of five critical regions is obtained for this mp-DO problem on application of the method presented in Section 2.2.1, as shown in Fig. 2.11. It turns out that this partition could in fact be extended to the entire right-half plane $\{\theta_1 \geq 0\}$. The critical region CR_1 corresponds to unconstrained optimal solutions. In CR_2 and CR_3, the solutions are comprised of two interior arcs separated by a touch-and-go point for the path constraints $x_1(t) \leq \theta_1$ and $x_1(t) \geq -\theta_1$, respectively, a behaviour characteristic of higher-order path constraints; this behaviour is illustrated for the optimal control solution for $\theta = (0.15, 0.8) \in \mathrm{CR}_2$ on Fig. 2.12(a), where the terminal constraint $x_1(1) \leq 0$ is also seen to be active. Finally, in CR_4 and CR_5, the optimal solutions are comprised of three arcs, the first and last ones unconstrained and the middle one binding for the path constraints $x_1(t) \leq \theta_1$ and $x_1(t) \geq -\theta_1$, respectively; this behaviour is illustrated for the optimal control solution for $\theta = (0.1, 0.8) \in \mathsf{CR}_4$ on Fig. 2.12(b), where the path constraint $x_1(t) \leq 0.1$ remains active along the time interval $[0.375, 0.625]$.

2.4.3 Multi-parametric NCO-tracking control of an FCC unit

This final case study considers a fluidized catalytic cracking (FCC) unit operated in partial combustion mode [85]. The objective is to steer the system to a given operating point, defined in terms of the mass fraction of coke on regenerated catalyst, C_{rc}, and the regenerator dense bed temperature, T_{rg}. The manipulated variables are the flow rate of air sent to the regenerator, F_{a}, and the catalyst flow rate, F_{s}. A linear input-output dynamic model is obtained via linearization and reduction of a first-principles nonlinear model around the equilibrium point $C^*_{\mathrm{rc}} = 5.207 \times 10^{-3}$, $T^*_{\mathrm{rg}} = 965.4\,\mathrm{K}$, $T^*_{\mathrm{ro}} = 776.9\,\mathrm{K}$, $T^*_{\mathrm{cy}} = 988.1\,\mathrm{K}$, and $T^*_{\mathrm{f}} = 400\,\mathrm{K}$, where T_{f} denotes the feed oil temperature. The control and state variables in this linear dynamic system are

$$x(t) := \begin{bmatrix} C_{\mathrm{rc}}(t) - C^*_{\mathrm{rc}} \\ T_{\mathrm{rg}}(t) - T^*_{\mathrm{rg}} \end{bmatrix}, \quad \text{and} \quad u(t) := \begin{bmatrix} F_{\mathrm{s}}(t) - F^*_{\mathrm{s}} \\ F_{\mathrm{a}}(t) - F^*_{\mathrm{a}} \end{bmatrix}.$$

The optimization-based control problem is given by

$$
\begin{aligned}
\min_{u(t),x(t),y(t)} \quad & \frac{1}{2}x(t_{\mathrm{f}})^{\mathrm{T}}Q_{\mathrm{f}}x(t_{\mathrm{f}}) + \int_0^{10} \frac{1}{2}x(t)^{\mathrm{T}}Qx(t) + \frac{1}{2}u(t)^{\mathrm{T}}Ru(t)\,dt \\
s.t. \quad & \dot{x}(t) = F_x x(t) + F_u u(t) + F_\theta \theta_3 \\
& g(x(t), u(t)) \le 0 \\
& x(0) = (\theta_1, \theta_2)
\end{aligned}
\tag{2.33}
$$

with:

$$
\begin{aligned}
Q_{\mathrm{f}} &= \begin{bmatrix} 3.011\times 10^{7} & 1334 \\ 1334 & 1.260 \end{bmatrix}, \quad Q = \begin{bmatrix} 10^{8} & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
F_x &= \begin{bmatrix} -2.55\times 10^{-2} & 1.51\times 10^{-6} \\ 227 & -4.10\times 10^{-2} \end{bmatrix}, F_u = \begin{bmatrix} 3.29\times 10^{-6} & -2.60\times 10^{-5} \\ -2.8\times 10^{-2} & 7.80\times 10^{-1} \end{bmatrix}, \\
F_\theta &= \begin{bmatrix} 6.87\times 10^{-7} \\ 2.47\times 10^{-2} \end{bmatrix},
\end{aligned}
\tag{2.34}
$$

and the path constraints $g(x(t), u(t)) \le 0$ given by:

$$
\forall t \in [0, 10], \quad \begin{bmatrix} -100 \\ -15 \end{bmatrix} \le u(t) \le \begin{bmatrix} 100 \\ 15 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -10^{-3} \\ -20 \end{bmatrix} \le x(t) \le \begin{bmatrix} 10^{-3} \\ 20 \end{bmatrix}.
$$

The parameters θ in this mp-DO formulation correspond to uncertainty in C_{rc}, T_{rg}, and T_{f}.

We solve the mp-DO problem (2.33) with the approach presented in Section 2.2.1, for the uncertainty set $\Theta := [-10^{-3}, 10^{-3}] \times [-20, 20] \times [-10, 10]$. Overall, 11 critical regions are obtained, as shown in Fig. 2.13 – both in a 3-d plot (Fig. 2.13(a)) and the corresponding 2-d projections for the parameter value $\theta_3 = 0$ (Fig. 2.13(b)); the discretized mp-QP counterparts for the latter will be presented later on (Fig. 2.15).

- The critical region CR_1 corresponds to unconstrained optimal controls; for instance, the optimal solution shown in Fig. 2.14(a) with $\theta^{(1)} = (5\times 10^{-4}, 5, 0) \in \mathsf{CR}_1$.
- The solutions in both CR_2 and CR_7 are comprised of two arcs, a boundary arc where u_2 reaches its lower bound or its upper bound, respectively, followed by an interior arc. This situation is illustrated in Fig. 2.14(b) for $\theta^{(2)} = (10^{-4}, 20, 0) \in \mathsf{CR}_2$.

- The solutions in both CR_6 and CR_{11} are comprised of three arcs, with a boundary arc where x_1 reaches its upper bound or its lower bound, respectively, located in between an initial interior arc and a final one. In both CR_5 and CR_{10}, the solutions have the same constrained arc as CR_6 and CR_{11}, respectively, yet without the final interior arc as the state constraint remains active until the terminal time.

- The solutions in CR_3 and CR_8 combine the previous two cases in CR_2+CR_6 or CR_7+CR_{11}, and give rise to four arcs, starting with a boundary arc for u_2 (as in CR_2 or CR_7), followed by an interior arc, a boundary arc for x_1 (as in CR_6 or CR_{11}), and another interior arc; this case is illustrated in Fig. 2.14(c) for $\theta^{(3)} = (6 \times 10^{-4}, 20, 0) \in \mathsf{CR}_3$. The solutions in CR_4 and CR_9 present the same structure as CR_3 and CR_{11}, respectively, but lack the final interior arc after the boundary arc where x_1 is active.

For comparison, the critical regions shown in Fig. 2.15 are for the discretized mp-QP counterparts of (2.33) with either $N = 20$ or $N = 50$ time subintervals and for $\theta_3 = 0$. In both cases, the central regions correspond to unconstrained solutions, like CR_1 in Fig. 2.13; all the other critical regions contain one or more active constraints along certain time subintervals, thereby approximating the six critical regions CR_2-CR_{11} in Fig. 2.13 and the nonlinear feedback control laws thereof in terms of affine control laws only. The actual switching structure gets better approximated as the time discretization is refined, but this also leads to a significant increase in the number of critical regions, namely 175 with $N = 20$ and 1687 with $N = 50$. In sum, the comparison with a classical mp-QP approach confirms the clear benefit offered by a continuous-time mp-DO approach in terms of a lesser number of critical regions, along with the ability to capture the underlying nonlinear feedback control nature.

Closed-loop responses of the FCC unit based on the multi-parametric NCO-tracking controller derived from mp-DO problem (2.33) are shown in Fig. 2.16, starting from the uncertainty scenario $(8 \times 10^{-4}, 15, 0) \in \mathsf{CR}_6$ at $t = 0$. A sampling period of $\Delta T = 10^{-2}$ and signal-to-noise ratios of 50dB (Gaussian white noise) are considered in both cases, yet with different scenarios for the feed temperature. In the left plot (Fig. 2.16(a)), a nominal feed temperature $T_{/rmf}$ is used, so $\theta_3 = 0$ at all times; whereas the right plot

(Fig. 2.16(b) correspond to multiple step changes in the value of θ_3 as

$$\theta_3 = \begin{cases} 0 & \text{if } 0 \leq t \leq 5 \\ 5 & \text{if } 5 \leq t \leq 10 \\ -5 & \text{if } 10 \leq t \leq 15 \\ 0 & \text{if } t \geq 15\,. \end{cases}$$

The resulting control and response trajectories (solid lines) present a good control performance compared to the nominal case (dashed lines: noise-free and infinite sampling).

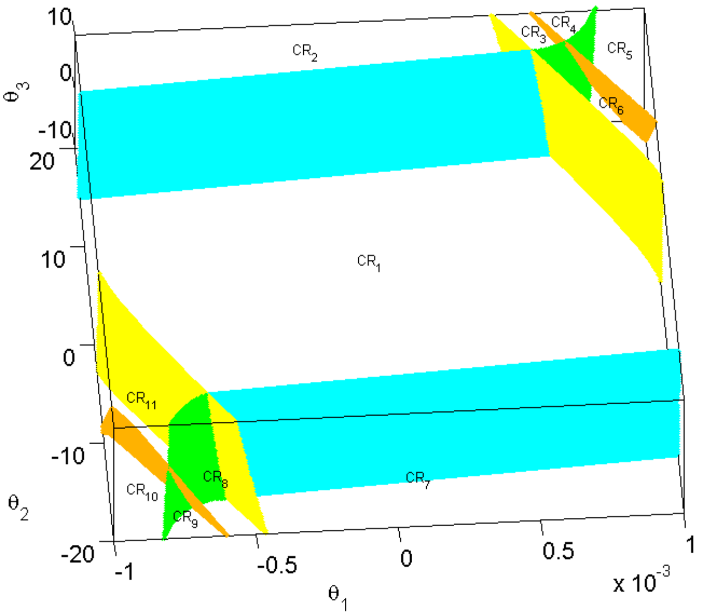
2.5 Conclusions

This chapter has introduced a framework for multi-parametric NCO-tracking that exploits the multi-parametric solution structure of an uncertain optimal control problem, without the need for applying a time discretization. The methodology involves two steps, the first off-line step that defines the multi-parametric control switching structure based on solving mp-DO problem, which results in a partitioning of the uncertain parameter domain into a number of critical regions, each corresponding to a unique sequence of active path constraints or active terminal constraints; the second on-line step involves locating the parameter into critical regions and applying the corresponding feedback laws.

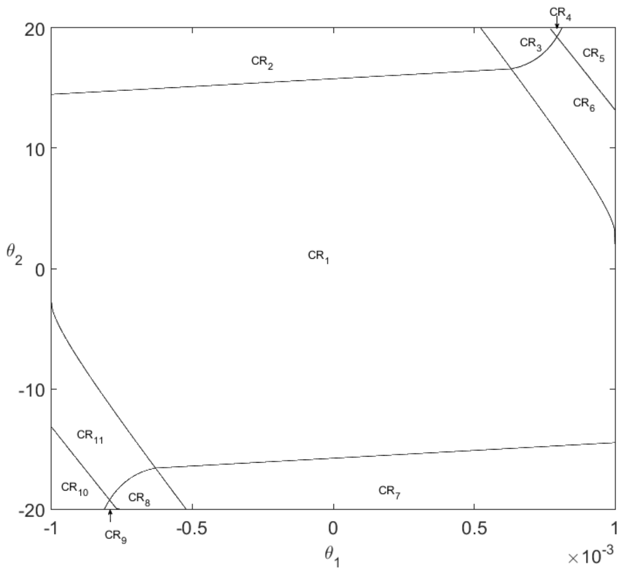
In the the special case of linear-quadratic optimal control problems, an algorithm has been proposed for characterizing the corresponding multi-parametric solution structure in terms of the exact critical regions and nonlinear feedback control laws. In practice, these feedback laws can be applied in a receding horizon manner, effectively resulting in a closed-loop, multi-parametric NCO-tracking controller for the system.

In comparison to classical NCO-tracking, this approach no longer requires the assumption of an invariant active set in the presence of uncertainty and extends the scope of NCO tracking to receding horizon control; whereas addressing the uncertain optimal control problem in its native, continuous-time form may lead to a dramatic reduction in the number of critical region compared to the classical mp-QP approach based on time discretization, due to the ability to capture the underlying nonlinear feedback control nature.

These properties have been illustrated with several examples throughout the paper, including the two-input control problem of an FCC unit. The critical regions and their corresponding switching structures are shown for examples with different orders of constraints. In the closed-loop simulation of the final case study, the parameter switches between different critical regions, which corresponds to the update of the control switching structures.

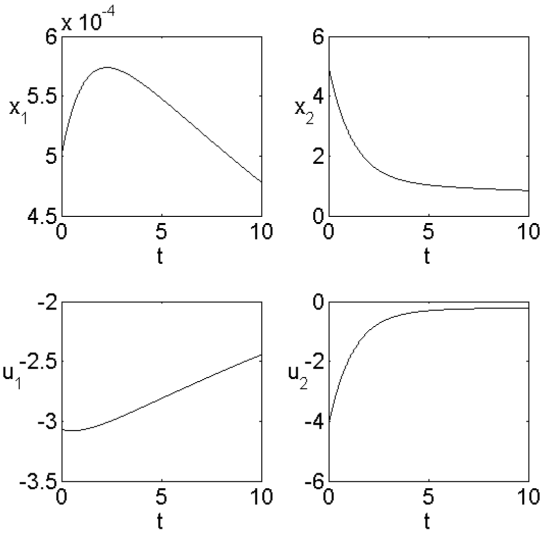


(a) Full parameter space

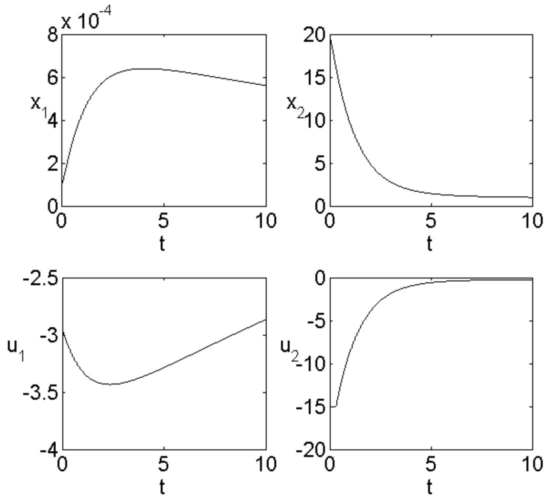


(b) Projections for $\theta_3 = 0$

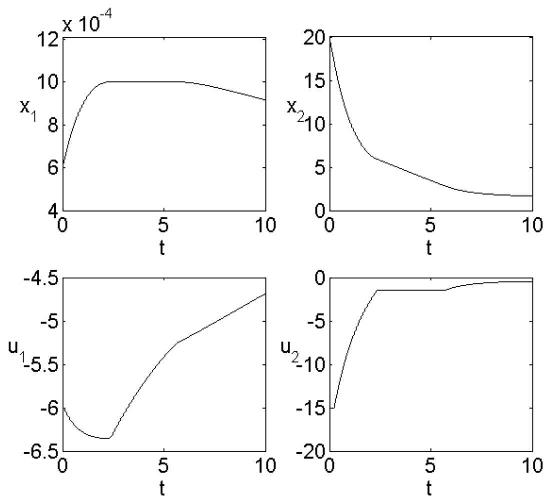
Figure 2.13: Critical regions of the mp-DO problem (2.33).



(a) $\theta^{(1)} = (5 \times 10^{-4}, 5, 0) \in \mathsf{CR}_1$

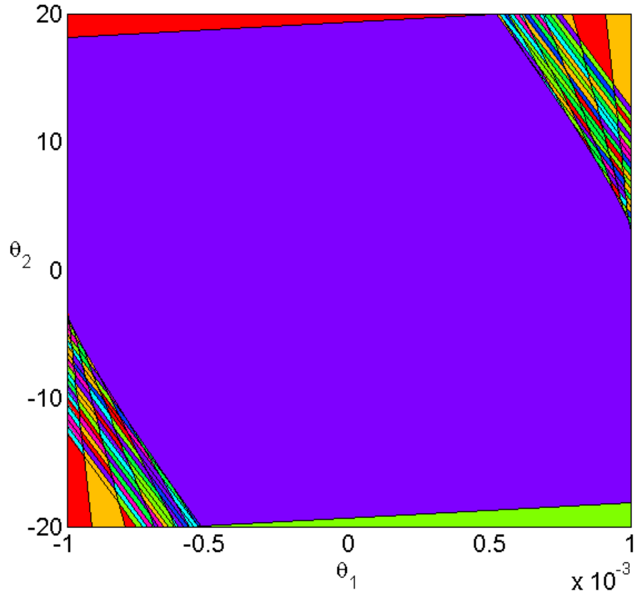


(b) $\theta^{(2)} = (10^{-4}, 20, 0) \in \mathsf{CR}_2$

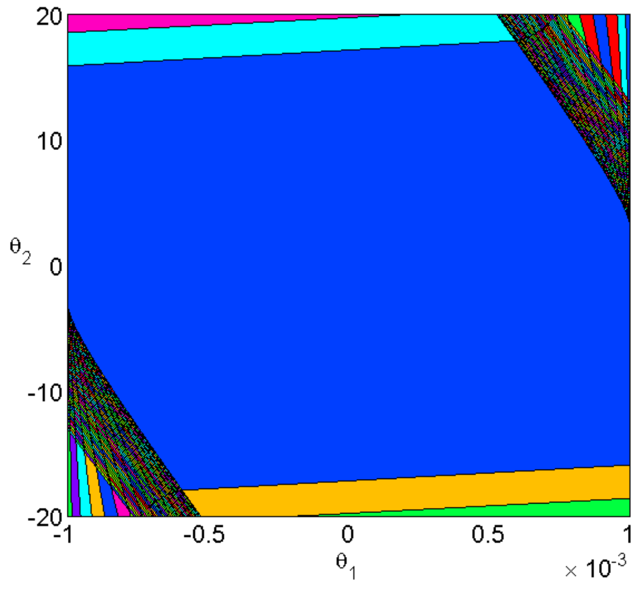


(c) $\theta^{(3)} = (6 \times 10^{-4}, 20, 0) \in \mathsf{CR}_3$

Figure 2.14: Selected optimal control and response trajectories of the mp-DO problem (2.33).

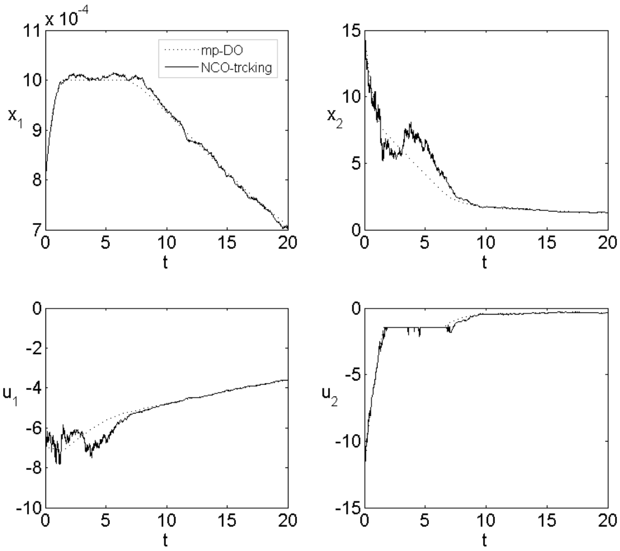


(a) $N = 20$

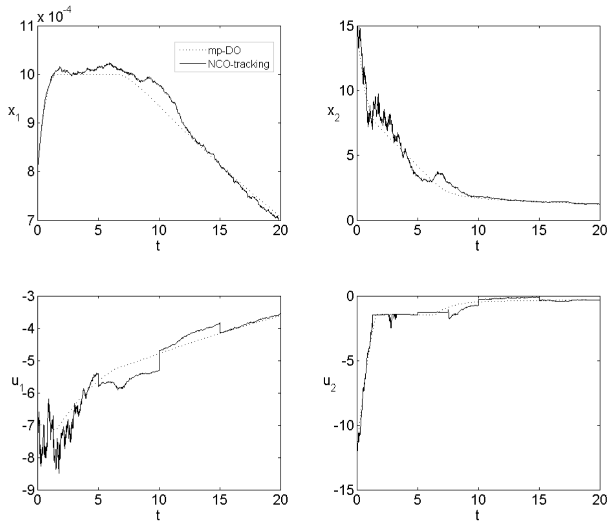


(b) $N = 50$

Figure 2.15: Critical regions of discretized mp-QP counterparts of problem (2.33) for $\theta_3 = 0$.



(a) $\theta_3 = 0$ all the time and 50dB signal-to-noise ratio



(b) θ_3 with step changes and 50dB signal-to-noise ratio

Figure 2.16: Closed-loop response of the multi-parametric NCO-tracking controller for problem (2.33).

Chapter 3

Multi-parametric NCO-tracking control based on data-driven classification

3.1 Introduction

This chapter presents a systematic procedure for characterizing the critical regions in continuous-time mp-DO problems and for searching a critical region during on-line execution. The critical regions in mp-DO may be non-convex and closed-form expressions describing their boundaries may not be available in general. This contrasts with explicit NLP, e.g. based on mp-QP, for which powerful detection and mapping of the critical regions are available, including geometrical techniques [35, 86, 87], combinatorics [88–90], and, more recently, graph-theoretic approaches [91, 92]. Herein, we investigate the use of data classifiers based on deep learning [93, 94] in order to characterize the critical regions in mp-DO. Such classifiers take the control problem parameters as inputs and map the corresponding critical regions in terms of their switching structures. Similar applications of machine learning within explicit MPC have been proposed for approximating the solution of both linear and nonlinear MPC [45, 95]. Another feature of data classification lies in its ability to estimate the likelihood of a given parameter value to belong within a certain critical region, thus providing a basis for the point-location problem during the on-line execution of mp-NCO-tracking.

An algorithm for training such classifiers is presented, involving building the classification model based on neural networks, generating training data and applying optimization based training for parameters of the network. We first introduce the general classification category, and then present a step-wise procedure for training a classifier for the mp-NCO-tracking control. The architecture of the classification model based on neural networks is introduced first, from a single neuron to a layer-wise structure. Afterwards, details are given on the definitions of score function for mapping input data to class scores and loss function for quantification of the prediction, which is to be minimized in the following training process. The sample data are labelled with the corresponding switching structures, and softmax classifier is chosen as the output layer of the neural network that minimizes the cross entropy between the predictions and the ground truth labels. By training a classification model based on the labelled sample data corresponding to different switching structures, a classifier is obtained representing the critical regions as partition of the parameter space. Taking a measurement from the dynamic system as input, the classifier computes a probability distribution on labels corresponding to critical regions and also provides a way to handle misclassification. The resulting classifier is conveniently embedded into the mp-NCO-tracking controller for locating the parameters into their corresponding critical regions.

The rest of the chapter is organized as follows. Section 3.2 presents an overview of the general classification methods in the field of machine learning. Section 3.3 presents the algorithm of training classifiers for the implementation on mp-NCO-tracking control problem, including building a classification model with a layer-wise architecture of a fully connected deep neural network, generating sample data and performing the optimization based training. Two numerical case studies are given in Section 3.4 to illustrate the approach, including training neural network based classifiers and their applications in the on-line feedback control problem. Finally, Section 3.5 concludes the chapter.

3.2 Classification category

A classifier is defined as an algorithm that implements classification, especially in a concrete implementation that maps input data to a category. Classifying data is a common task in machine learning, and a model is trained based on sample data and performs as a predictor for a new input. The problem critical region partitions in this research belongs to supervised multiclass classification that produces a model from a labelled training set. A category of the common algorithms for multiclass classification problem includes [93]:

1. Neural networks: multilayer architecture of neural networks provides a natural extension to the multiclass problem by approximating function models with arbitrary complexity, where the output layer can have multiple neurons leading to multiclass classification.

2. Support vector machines: support vector machines are based upon maximizing the margin between classes, i.e. maximizing the minimum distance from the separating hyperplane to data. In the extension for multiclass from its binary counterpart, additional parameters and constraints are added to the optimization problem to handle the separation.

3. k-Nearest Neighbor: kNN is considered among the oldest non-parametric classification algorithms, where the distance from the example to every other training example is measured. The class label is chosen as the most represented among the k classes with smallest distances. It is space inefficient and expensive for predicting.

4. Naive Bayes: Naive Bayes is a successful classifier based upon the principle of maximum a posteriori, and it is naturally extensible to the case of multiple classes and was shown to perform well in spite of the underlying simplifying assumption of conditional independence.

5. Decision trees: the tree aims to infer a split the training data from available feature values, and an example is classified by following a path from the root node to a leaf node, by performing a test on some feature at each node.

Training a classifier is based on a data-driven approach, which relies on first accumulating a dataset of labelled sample points, which is referred to as a training set. The main steps for training a classifier include obtaining training set and pre-processing, setting up a model, training based on the training set, and finally, evaluation on the quality of the classifier for prediction, i.e. the match between the predicted and the ground truth labels for new data. It is common to use the accuracy, which measures the fraction of correct predictions, as an evaluation criterion. Moreover, the test time efficiency for on-line recognition is more important than the training efficiency off-line. Neural networks are expensive to train, but once the training is completed, it is very cheap to classify new data, which is much more desirable in practice. With high accuracies, it allow us to completely discard the training set once after the training process and only keep the learned parameters for quickly evaluating new examples.

3.3 Implementation on mp-NCO-tracking

A central, and particularly arduous, task for the application of the mp-NCO-tracking methodology is detecting all the critical regions of the mp-DO problem (3.1), and representing those regions in a form that can be easily exploited in the on-line point location problem. Because the critical regions are generally non-convex and lack a closed-form representation, classification methods from the fast-developing field of machine learning, where the task is to predict a discrete class for a given input, represent a powerful alternative. In the context of mp-NCO-tracking, multinomial classifiers [93] can provide the desired mapping between the control problem parameter θ and the set of possible switching structures $\{\mathcal{S}^1, \ldots, \mathcal{S}^{NC}\}$.

Our focus in this chapter is on multilayer perceptron (MLP) [94], a class of neural networks featuring multiple (hidden) layers and non-linear activation functions, which has the ability to distinguish data sets that are not linearly separable. The MLPs of interest comprise exactly NC neurons in their output layers, each one corresponding to the unique switching structure for some $\theta \in \Theta$. A natural objective for MLP training is minimizing misclassification with respect to the training data set. In particular, we use a so-called softmax function, whose output $\mathcal{P} := [p_1, \ldots, p_{NC}]$ is a vector representing the probability for each switching structure. The normalized probabilities returned by the softmax classifier can subsequently be exploited on-line in the point location problem, where it can guide the search for the correct switching structure and thereby avoid a naïve enumeration; see Step II in Section 3.3.1. Other loss functions which can only recognize the class with the highest score, such as support vector machines, are not suitable in this context, which is discussed in Section 3.3.2. The specifics of MLP training for their application in mp-NCO-tracking are detailed in Section 3.3.3. Although the focus is on MLP and softmax classifiers, the methodology could of course be transposed to other classification techniques.

First, we briefly review the definition of critical regions and the corresponding conditions needs to by satisfied for a certain solution structure in [96]. The main components for training MLP for classification of critical regions include building a model, which consists of setting up the layer-wise architecture of the network and preprocessing the labelled sample examples as training data, optimization based training process to obtain the corresponding parameters and finally the evaluation for generalized performance.

3.3.1 Review of critical regions

Before presenting the classifier, we provide an overview of mp-DO as shown in (3.1) and the critical regions for mp-NCO-tracking control introduced in Chapter. 2.

$$
\begin{aligned}
\Phi(\theta) \quad &:= \min_{\substack{u(t),x(t),\\ t_0\le t\le t_\mathrm{f}}} \frac{1}{2}x(t_\mathrm{f})^\mathsf{T} Q_\mathrm{f} x(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \frac{1}{2}x(t)^\mathsf{T} Q x(t) + \frac{1}{2}u(t)^\mathsf{T} R u(t)\, dt \\
\text{s.t.} \quad & \dot{x}(t) = f(x(t),u(t)) := F_x x(t) + F_u u(t) + F_\theta \theta + f_0 \\
& g(x(t),u(t)) := G_x x(t) + G_u u(t) + G_\theta \theta + g_0 \le 0 \\
& h(x(t_\mathrm{f})) := H_x x(t_\mathrm{f}) + H_\theta \theta + h_0 \le 0 \\
& x(t_0) = B_\theta \theta + b_0 \,,
\end{aligned}
\tag{3.1}
$$

where $x \in \mathbb{W}_{1,2}^{n_x}$ denotes the state response; $u \in \mathcal{U} := \{u \in \mathbb{L}_2^{n_u} \mid \forall t \in [t_0, t_\mathrm{f}], u(t) \in U\}$ with $U \in \mathbb{K}_\mathrm{C}^{n_u}$, the control input; $\theta \in \Theta$ with $\Theta \in \mathbb{K}_\mathrm{C}^{n_\theta}$, the parameter; $w \in \mathcal{W} := \{w \in \mathbb{L}_2^{n_w} \mid \forall t \in [t_0, t_\mathrm{f}], w(t) \in W\}$ with $W \in \mathbb{K}_\mathrm{C}^{n_w}$, the time-varying uncertainty; and $Q_\mathrm{f} \in \mathbb{S}_+^{n_x}$, $Q \in \mathbb{S}_+^{n_x}$, and $R \in \mathbb{S}_{++}^{n_u}$ are given weighting matrices.

Each critical regions $\Theta^i \subseteq \Theta$, $i = 1 \ldots NC$ of the mp-DO problem (3.1) is comprised of those parameter values $\theta \in \Theta^i$ whose corresponding optimal control solutions all share a common switching structure $\mathcal{S}^i$. An explicit parameterization of the optimal solutions in a critical region can be obtained by exploiting the equality conditions in the parametric NCOs, and comes in the form

$$
\forall \theta \in \Theta^i, \quad (u(t), x(t), \lambda(t), \mu(t), \nu, \pi) = \mathcal{K}^i\left(t, \theta, t_1(\theta), \ldots, t_{N_t^i - 1}(\theta)\right) . \tag{3.2}
$$

The remaining inequality conditions in the parametric NCOs yield an implicit description of the boundary of a critical region. The boundary between two critical regions thus corresponds to either an inactive terminal or path constraint activating, or an active terminal or path constraint inactivating.

The mp-NCO-tracking methodology proposed in Chapter. 2 proceeds in two steps:

(I) The off-line step defines the multi-parametric control structure, which entails a partitioning of the parameter domain Θ into NC critical regions, $\Theta^1 \cup \cdots \cup \Theta^{NC} \subseteq \Theta$, each corresponding to a unique switching structure $\mathcal{S}^1, \ldots, \mathcal{S}^{NC}$.The feedback laws in each critical region are furthermore expressed in the parameterized form (3.2). The partitions of the critical regions can be saved in a classifier by training sample data, where its input is the parameter and its output is the label of the region Θ^i. Details of the algorithm for training the classifier will be presented in the following sections.

(II) The on-line step applies the feedback laws (3.2) in a receding horizon manner. Each sampling time entails the location of the critical region Θ^i containing the current parameter values θ. For a given switching structure candidate $\mathcal{S}^i$, this step starts by computing the switching times $t_1(\theta), \ldots, t_{N_t^i-1}(\theta)$ so as to enforce continuity of the Hamiltonian function at these times, for instance by applying a Newton iteration. A verification that θ is contained within Θ^i then consists of a simple check that all of the primal and dual feasibility conditions are satisfied under the feedback law $\mathcal{K}^i$, which is implemented until next update.

3.3.2 Build classification model

This section focuses on building a neural network model as a classifier for critical regions. The input of the classifier is the system parameter θ, and the output is a vector with as many elements as the total number of critical regions denoting a normalized probability distribution, i.e. $\mathcal{P} := [p_1, \ldots, p_{NC}]$, and each element p_i representing the possibility of a region Θ^i to be the correct one. The dimension with the highest score is taken as the first choice of predicted class label corresponding to the input parameter. When the input lies around the region boundaries, the class label with highest probability in output layer might not be correct. In the case of such a misclassification, a second choice with successive highest probability is readily available as the successive choice, and so on for the following ones. This approach provides a way for the recognition of the ground truth label based on a probability distribution.

3.3.2.1 Architecture of model

A M-P neuron model receives input signals from other n neurons, which are passed by weighted connections, and the total input is passed into an activation function for generating the output from the neuron. Mathematically, a neuron computes a dot product and adds a bias, followed by a nonlinear activation function as output, and it is arranged into layers in a neural network.

The output of a single neuron can be written in the following form:

$$y_i(x) = f(Wx + b_i) \tag{3.3}$$

where $x \in \mathbb{R}^n$ denotes the input signals; $W \in \mathbb{R}^{1\times n}$ is the weighting row vector and b_i is a bias scalar. The activation function $f(\cdot)$ then takes the single value y and performs certain mathematical operation on it. It is usually a simple continuous and differentiable function, such as Sigmoid: $\sigma(z) = 1/(1+e^{-z})$, which squashes a real-valued input to the range [0,1], but may saturate and kill gradients or introduce undesirable zig-zagging dynamics in the gradient updates for parameters; Tanh: $\tanh(z) = 2\sigma(2z) - 1$, which squashes a real-valued input to the range [-1, 1]; ReLU: $f(z) = \max(0, z)$, where the activation is simply threshold at zero; Leaky ReLU: $f(z) = \mathbb{I}(z < 0)(\alpha z) + \mathbb{I}(z >= 0)(z)$; Maxout: $\max(w_1^T x + b_1, w_2^T x + b_2)$, as a generalization of both ReLU and Leaky ReLU. It is very rare to mix different types of neurons in the same network, even though there is no fundamental problem with that.

Note that, a single neuron can work as a linear classifier, with the capacity of activation near one as true or near zero as false for certain linear regions and an loss function. It can be tuned into a binary softmax classifier (also known as logistic regression) with Sigmoid as activation function or a binary SVM with a max-margin hinge loss.

The neural networks considered in this chapter are modelled as collections of neurons that are connected in an acyclic graph and organized into distinct layers of neurons. For regular neural networks, the most common layer type is the fully-connected (FC) layer, where neurons between two adjacent layers are fully pairwise connected, while neurons within a single layer share no connections. The architecture of a fully connected multi-layer feed-forward neural network is shown in Fig. 3.1, consisting of an input layer I with

n neurons, two hidden layers H each with i neurons and an output layer P with NC neurons, which are used to represent the final class scores for classification.

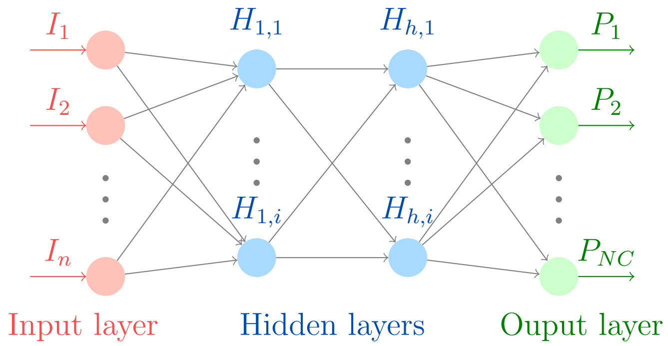


Figure 3.1: Layer-wise organization of a fully connected neural network

The forward pass of a fully-connected layer corresponds to one matrix multiplication followed by a bias offset and an activation function evaluation. For a layer with m neurons, the output of the layer is a vector obtained by a linear mapping combined with activation functions,

$$y_{out} = f(Wz + b) \tag{3.4}$$

where $z \in \mathbb{R}^{\mathrm{n}}$ is the output vector from its previous layer; each row of the weight matrix $W \in \mathbb{R}^{\mathrm{m \times n}}$ is associated with a single neuron and $b \in \mathbb{R}^{\mathrm{m}}$ is a bias vector for this layer. The weighting matrix and a bias vector, which are free parameters to be chosen in training process such that the correct class has a score higher than the scores of incorrect classes. Together, these choices define the form of the score function, which is extended from simple linear mappings to arbitrary complex nonlinear computations by introducing layer-wise architectures..

This makes it simple and efficient (parallel for all training data) to evaluate using matrix vector operations for neural networks with a layer-wise structure. The neurons in the output layer most commonly do not have an activation function, because the last output layer is usually taken to represent the class scores in classification or some real-valued target in regression. Overall, a neural network performs a sequence of linear mappings with interwoven nonlinearities.

From the mathematical point of view, neural networks with fully-connected layers can be regarded as a family of functions organized in a layer-wise connected structure that are parameterized by the connection weights. Neural network with at least one hidden layer are universal approximators for any continuous functions [94, 97]. Given any continuous function $f(x)$ and some $\epsilon > 0$, there exists a neural network $g(x)$ with one hidden layer with a reasonable choice of non-linearity such that $\forall x, |f(x) - g(x)| < \epsilon$, i.e. the neural network can approximate arbitrary continuous functions.

Although a two-layer neural network can be a universal approximator, the fact that deeper networks with multiple hidden layers can work better than a single-hidden-layer networks is an empirical observation, despite the fact that their representational powers are equal. The capacity of the network increases with the size and number of layers, and together the space of representable functions grows. Since the neurons can collaborate to express different functions, larger neural networks can represent more complicated functions. Increasing the depth of network is more powerful, since it also increases the cascade layers of function embeddings. Although the capacity of a model increases as its complexity increases, an excessively complex network might come cross the problem of overfitting. One should use as big of a neural network as the computational budget allows, and use other regularization techniques to control overfitting coming from noises in the data for better generalization in prediction. In addition, the development in cloud computing and big data has made the computing ability been dramatically improved, and the risk of overfitting can be reduced by increasing the training data size.

After the architecture of the network is chosen, the parameters need to be initialized before training process, commonly by using small random numbers from a zero mean, unit standard deviation Gaussian, further calibrating the variances of the weights and batch normalization [98–100]. There are many hyperparameters to define for training the neural networks, including the initial learning rate, learning rate decay schedule and regularization [101]. The sizes of the intermediate hidden layers are also hyperparameters of the network, and random search often give better performance than grid search [101].

3.3.2.2 Loss function definition

For the supervised classification problem considered in this chapter, training data are given and fixed, and the optimization procedure is to minimize the loss function subject to the parameters of a neural network based on the labelled data set. A classifier has two major components, a score function for computing the class scores and a loss function that quantifies the agreement between the predicted scores and the ground truth labels, which can be treated as the cost function of an optimization problem with respect to the adjustable parameters in the score function. Thus, a better prediction is equivalent to a smaller loss function.

To define the loss function, we first define a data set of sample points $\{x_i\} \subset \mathbb{R}^{\mathrm{D}},\ i \in [1, N]$ and each associated with a ground-truth label $y_i \in [1, K]$. Let $s(x_i)$ denote the class scores vector for x_i, where $s_j(x_i)$ is the score of the j-th class for x_i. The score function $f : \mathbb{R}^D \to \mathbb{R}^K$ is the mapping from data to class scores by forward passing through passing through all connected layers. Finally, a score for each class is obtained as a function of all the weights and bias vectors as follows:

$$s(x_i) = f(x_i, W_1, \ldots, W_N, b_1, \ldots, b_N) \tag{3.5}$$

Afterwards, the loss function representation is chosen to character the match between prediction and the ground truth label, and two most common choices are support vector machine and softmax classifiers, which are introduced as follows:

(a) Support vector machines (SVMs) aim to find the maximum-margin hyperplanes that separate two sets, and the resulting classifier is also known as maximum-margin classifier [102]. Multiclass SVMs are designed to make the score of the correct class higher than all the other scores by at least a margin of delta. If any class has a core higher than the lower bound, there will be accumulated loss, otherwise the loss is zero. The objective of optimizing the model is to simultaneously satisfy the constraints for the training data set and minimize the total loss.

The multiclass SVM loss for x_i is defined as [103]

$$L_i = \sum_{j \neq y_i} \max(0, s_j(x_i) - s_{y_i}(x_i) + \Delta) \tag{3.6}$$

By a substitution of the linear score function (3.5), L_i can be written as

$$L_i = \sum_{j \neq y_i} \max(0, W_{(j,:)}^T x_i - W_{(y_i,:)}^T x_i + \Delta) \tag{3.7}$$

The threshold at zero $\max(0, -)$ is called hinge loss, and it is very common to use the form $\max(0, -)^2$ that penalizes violated margins more strongly, which is referred to as L2-SVM. Nonlinear classifiers can be developed by applying kernel tricks to the maximum-margin hyperplanes, where each dot product is replaced by a nonlinear kernel function [102]. Other commonly used forms for multiclass SVM include one-vs-all, lone-vs-one, or all-vs-all strategies, which are based on training binary SVM.

(b) Softmax classifier is a generalization to multiple classes based on the binary logistic regression classifier. By interpreting the score functions (3.5) as normalized log probabilities for each class, the cross-entropy loss takes the form as

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right) = -s_{y_i} + \log\sum_j e^{s_j} \tag{3.8}$$

where s_j denotes the j-th element of class scores s; y_i denotes the correct label corresponding to x_i.

The softmax function

$$f_k(s) = \frac{e^{s_k}}{\sum_j e^{s_j}} \tag{3.9}$$

takes the vector of class scores s and squashes it to a vector of values between zero and one that sum to one, thus generating an output as normalized class probabilities. From the information theory point of view, the cross-entropy between a true distribution p and an estimated distribution q

$$H(p, q) = -\sum_x p(x)\log q(x) \tag{3.10}$$

is minimized by a Softmax classifier based on the estimated class probabilities ($q_i = \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$) and the true distribution of classes ($p_i = [0, \ldots, 1, \ldots, 0]$ with 1 at y_i-th position). From the probabilistic point of view,

$$P(y_i|x_i) = \frac{e^{s_{y_i}}}{\sum_j e^{s_j}} \tag{3.11}$$

can be interpreted as the normalized probability assigned to the correct label y_i given the input data x_i, i.e. the optimization of classifier aims to minimize the negative log-likelihood of the correct class, as performing maximum likelihood estimation.

For numeric stability, a normalization is usually applied as follows:

$$\frac{e^{s_{y_i}}}{\sum_j e^{s_j}} = \frac{Ce^{s_{y_i}}}{C\sum_j e^{s_j}} = \frac{e^{s_{y_i}+\log C}}{\sum_j e^{s_j+\log C}} \tag{3.12}$$

where the value of C is usually chose as $\log C = -\max_j s_j$, which is equivalent as to shift the class score by subtracting the highest value.

The full loss function J to be minimized consists of the data loss L and a regularization term R as follows:

$$J = L + \alpha R(W) \tag{3.13}$$

where the regularization penalty is appended to the loss function weighted by hyper-parameter α, which is usually obtained by hyper-parameter tuning. Note that, due to the regularization penalty, the loss function can never be exactly zero.

The data loss function is defined over all individual data loss L_i for each sample point x_i as follows:

$$L = \frac{1}{N}\sum_i L_i \tag{3.14}$$

Regularization term is added for controlling the capacity of neural networks, such as L1/L2 regularization, max norm constraints or Dropout, but it should not be too large to overwhelm the data loss [104]. In practise, it is most common to use a single global L2 regularization as follows:

$$R(W) = \sum_k \sum_n W_{k,n}^2 \tag{3.15}$$

which prefers smaller and more diffuse weight vectors. Thus, the classifiers are encouraged to take into account all input dimensions with small amounts rather than a few input dimensions strongly, which helps to improve the generalization performance and lead to less overfitting [105]. The regularization does not include bias b, so the influence of an input dimension is not controlled.

The full loss of a SVM with L2 regularization penalty can be written as

$$L = \frac{1}{N}\sum_i \sum_{j \neq y_i} \max(0, s_j(x_i) - s_{y_i}(x_i) + \Delta) + \alpha \sum_k \sum_n W_{k,n}^2 \tag{3.16}$$

where the hyper-parameters Δ and α control the same trade-off between the data loss and the regularization loss in the objective, and it is safe to always set $\Delta = 1.0$.

The full loss of a Softmax classifier with L2 regularization penalty can be written as

$$L = \frac{1}{N}\sum_i (-s_{y_i} + \log \sum_j e^{s_j}) + \alpha \sum_k \sum_n W_{k,n}^2 \tag{3.17}$$

where the regularization term can be interpreted as coming from a Gaussian prior over the weight matrix W, as performing maximum a posteriori estimation.

It can be seen that SVM uses the max-margin loss to compute uncalibrated and not easy to interpret scores for all classes, while softmax uses the cross-entropy loss to compute probabilities for all labels, which can be interpreted as the confidence in each class. But the diffusion of these probabilities also depends on the regularization penalty, such that the same unnormalized log-probabilities distributions may lead to different normalized outputs by different regularization strength α. In practice, SVM and softmax are usually comparable, the performance difference are usually very small. However, SVM does not consider the details of the individual scores, while softmax classifier is always trying to assign correct class a higher probability and the incorrect classes lower probabilities, such that the loss is always getting smaller.

Therefore, softmax classifier is a suitable choice for the classification problem concerned in this research. A probability distribution for potential switching structures not only predicts the class with the highest score, but also gives a rank for all the other classes to handle the case of misclassification. In fact, due to the continuity of the parameter space and no gap existing between adjacent regions, sample points belongs to different regions may position very close in the boundary region. And it might happen for a point that the class scores for its neighbouring regions are higher than that for the correct region. In case of a point is misclassified by the first choice, successive choices can be made to obtain a correct label.

3.3.2.3 Generating sample data

Generating labelled sample data is a key step for the supervised classification problem. In this research, the labeling of samples calls for detecting the switching structure of the optimal control solution corresponding to every sample $\theta^{(1)}, \ldots, \theta^{(M)}$. This detection can be made from the numerical solution of the optimal control problem (3.1). On parameterizing the control trajectory, $u(t) = \phi(\nu)$, where ν is a finite set of parameters and the parameterization ϕ corresponds to a piecewise constant or linear function, and discretizing the path constraint as interior-point constraints, one can approximate the infinite-dimensional problem in (3.1) with a convex quadratic program (QP) for which fast and reliable solvers are available. The active constraints in the QP solution correspond to active path constraints at given times and active terminal constraints. The structure of the actual optimal control can then be inferred by inspection of the KKT multipliers and collecting the adjacent active times for the path constraints, as described in [39]. Labels for sample data is defined such that each label corresponds to a unique switching structure $\mathcal{S}^k$, the target (ground-truth output) of the network is then $y(\theta) = [0, \ldots, 1, \ldots, 0]$ with the k-th element to be 1 and others 0.

Sampling is a very important aspect for building an effective and economical data generation. For the efficiency of global representation over the entire parameter space, an essential consideration is the trade-off between the exploration and exploitation [106, 107]. The former focuses on distribution of the data points over the complete domain so as to fill up and spread as evenly as possible. While exploitation favours the regions that are identified as potentially interesting, thus to make larger information gain. Since the distribution information for critical regions are not known, sample data collection should be selected iteratively and guided by previously evaluated samples and their output values, which calls for the process of adaptive sampling, also know as active learning.

Adaptive sampling avoids generating superfluous data by putting emphasis on the most important boundary areas with more variance, which belongs to sequential sampling designs [108–110]. For the classification problem for critical regions, the areas with high variance represent the boundaries of the critical regions, discontinuity only happens when crossing the boundaries and all the interior points of a critical region share the same output value. Thus, the information gain is much higher in the boundary regions where higher density sampling should be generated.

In the training process, we use adaptive sampling strategy to improve the training efficiency and accuracy, such that more sample points are generated in the areas with higher variance or nonlinearities. The initial sampling can be performed based on some space filling techniques, such as Sobol or Latin Hypercube sampling that cover the feasible region as uniformly as possible [111, 112]. Afterwards, more data points are added in a sequential way partially based on the importance of the local regions, using outputs calculated from intermediate model as prior information to guide the sampling process. After each iteration, the points with incorrect prediction by the neural work or with a maximum value in output elements under certain level are labelled as reference points for finer sampling. The neighbours of these reference points denote more uncertain areas or regions with low confidence, and more sample points in these areas are added to the training set. With this adaptive sampling strategies, the total number of samples can be reduced substantially.

Moreover, to account for the exploration aspect, independent random points in the entire domain are added for increasing the global density and randomness. For the balance of the structure, the number of data for each class is also monitored, if it is too less, extra data points around the existing samples of that class are generated and used for training the network. Although not generating the exactly boundary points, the biased sampling strategies help to improve the information gain towards the more interested regions and to improve the efficiency to achieve the desired accuracy. At the same time, the distributed computing could be possibly applied for computations in sample data generations and the following training process [113].

After labelled samples are generated, data preprocessing is an important step before fitting them into an optimizer for training, and it is a very common practice to always perform normalization of input features to approximately the same scale for each dimension. Besides, data set is usually divided into training set, validation set and testing set. When the lack of training data is a concern, cross-validation can be used to help estimate the hyper-parameters. Here, the testing set is generated independently from the training set and is only used for evaluation of generalized performance of the model.

3.3.3 Training model

Training a neural network model aims to minimize the loss function that measures the quality of a particular set of parameters based on agreement between the predicted class scores and their ground truth counterparts. It can be considered as an optimization problem with loss function as the objective function with respect to the parameters of the score function, i.e. W and b involved in (3.13). After training process for the parameters is completed, only the matrix W and the vector b are stored and all the training set can be discarded. For computing a prediction, the new data is forwarded through the score function and classified based on the class scores.

3.3.3.1 Optimization based learning

The learning procedure aims to find the the optimal values of the parameters and hyper-parameters that minimize the loss function. Although training a neural network is generally a large-scale nonlinear optimization problem and there may exist lots of local minima, practical implementations show that the performance of some local solutions are good enough for the demanding tasks.

The optimization approaches for training neural networks is still currently a very active research area and the most common methods includes [101, 114–118]: first order gradient decent method, such as mini-batch gradient descent, stochastic gradient descent and momentum update; second order method based on Newton's method, such as quasi-Newton methods; adaptive learning rate methods, such as Adagrad, RMSprop and Adam. Furthermore, the quantities that should be monitored during learning process usually include loss function, validation/training accuracy (insights into overfitting) and ratio of update magnitudes to the value magnitudes.

Gradient descent method is the most common way at the core of all neural network libraries, which repetitively evaluates the gradient and preforms a parameter update as

$$p \leftarrow p + \Delta p \tag{3.18}$$

The step size for update is called learning rate, one of the most important hyper-parameters settings in training a neural network. During the procedure of parameter update, decaying the learning rate over the period of the training helps to stabilize the system iterations. In practice, only gradient for the parameters are usually computed, but gradient on training data can still be useful sometimes for the purposes of visualization or interpreting the task of neural network. Gradient can be calculated numerically or analytically, and a gradient check is usually performed for analytic gradient compared with numerical gradient by relative error. Since the loss functions are generally non-differentiable, subgradient is also widely used instead.

Automatic differentiation efficiently computes the gradients on the connections of the neural network with respect to the loss function [119, 120], where gradient can be calculated analytically using the chain rule for efficiently optimizing relatively arbitrary loss functions that express all kinks of neural networks. Based on the the back accumulation on local gradient recursively, any compound expression can be decomposed into simple interpretations by breaking down the forward pass into stages that are easily back-propped through. After the forward pass is finished, a gate eventually learns about the gradient of its output value on the final output of the entire network during backpropagation. Breaking down the forward pass into stages that are easily backpropped through is referred to as stages backpropagation. Each stage can be differentiated independently and expressions of the gates should have easy local gradients and can be chained together with least amount of efforts. Any kind of differentiable function can act as a gate, and multiple gates can be grouped into a sing gate or one function can be decomposed into multiple gates whenever it is convenient. The most commonly used patterns of gates in neural networks are add gate that distributes gradient of output equally to all inputs), max gate that equals 1.0 for highest input and 0 for others, and multiply gate that swaps inputs for multiplication with gradient of output. Magnitude of the gradient of the parameters depends on the scale of the data, such as multiply gate Wx with W as weights, thus data preprocessing is important for training a network.

3.3.3.2 Evaluation of accuracy

Since the objective of the classifier concerned in this research is to predict the correct class label for a given measurement, the performance of a network is referred to the accuracy on prediction. For a sample set $\mathcal{D} = \{(x_i, y_i)\}, i \in [1, n]$, by comparing the prediction $\hat{y}(x_i)$

with the ground-truth label y_i, the generalization error and accuracy rate are defined as follows:

$$\begin{aligned} &\text{Error rate:} \qquad E(\hat{y};\mathcal{D}) &&= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(\hat{y}(x_i) \neq y_i) \\ &\text{Accuracy:} \qquad Acc(\hat{y};\mathcal{D}) &&= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(\hat{y}(x_i) = y_i) = 1 - E(f;\mathcal{D}) \end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

For a classifier with an error rate ϵ, the probability that m out of n samples are misclassified is

$$P(m|\epsilon,n) = \begin{pmatrix} n \\ m \end{pmatrix} \epsilon^m (1-\epsilon)^{n-m}\,, \tag{3.19}$$

which is the probability of obtaining a testing error rate $\hat{\epsilon} = m/n$, and it satisfies the Binomial distribution. Thus, binomial test can be used to estimate the true error of classifier from the testing error rate. If the testing error rate $\hat{\epsilon} < \bar{\epsilon}$, such that

$$\bar{\epsilon} = \max \epsilon \qquad s.t. \sum_{m=\epsilon_0 \times n+1}^{n} \begin{pmatrix} n \\ m \end{pmatrix} \epsilon^m (1-\epsilon)^{n-m} < \alpha\,, \tag{3.20}$$

the assumption of $\epsilon < \epsilon_0$ is accepted with confidence $1-\alpha$. Thus, given the testing error rate $\hat{\epsilon}$ and the sample size, an estimation of the generalized prediction accuracy with confidence can be obtained based on the inverse cumulative density function, such that $\epsilon_0 = \epsilon_0(\hat{\epsilon}, \alpha, n)$. The higher the confidence, the lower the estimated accuracy is obtained.

The criterion for stopping the MLP training compares an estimate of the true classifier error rate derived from the binomial distribution, with the actual misclassification rate in the testing set. The confidence level $1-\alpha$ used in this hypothesis testing is usually set to 95%. Notice that such a termination criterion for the MLP training ensures that the test is statistically significant provided the testing set is sufficiently large, but it does not provide any guarantee that the trained MLP will describe the mp-DO critical regions to a desired accuracy. The ability of an MLP to describe a large number of nonconvex critical regions can only improve with an increasing number of hidden layers and number of neurons in these layers. The caveat is of course that a greater (off-line) computational burden is necessary for training a bigger MLP and for labelling the samples in a larger training set $\mathcal{T}$.

3.3.3.3 Algorithm

The off-line training of an MLP requires a labelled training set, learning from this training set, and checking the evaluation accuracy against a certain reference, possibly within an iteration loop. This overall procedure is summarized in Algorithm 4.1 below. The outcome is a trained MLP, which encapsulates a compact representation of the critical regions for the mp-DO problem at hand, and can be passed to the mp-NCO-tracking controller for on-line use.

A few remarks are in order. The initial sample set in Step 1 can be generated with a space filling technique, such as Sobol or Latin Hypercube sampling. Setting up the network includes choosing the layer structure and functions for neurons, as well as training data preprocessing, such as normalization. The learning process in Step ii aims to minimize a softmax function representing the mismatch between the predicted and actual switching structures across the entire training set. Here, Adam algorithm is used for optimization of the parameters in the neural networks in the framework of TensorFlow [121, 122], while there exist lots of additional resources for construction and optimization of the networks. A regularization term is normally added to this softmax criterion, e.g. the sum of squared weights in the MLP, in order to diffuse the weights throughout the MLP neurons and prevent over-fitting. The generalized performance of the classifier is computed based on (3.20), where each pair of the desired true error rate with a confidence level corresponds to a testing error rate over an independent testing set of certain size. Thus testing accuracy can be used directly as the stopping criterion, and the estimation on the true error rate is calculated by using the inverted cumulative density function of Binomial distribution.

Lastly, the aim of the adaptive sampling strategy in Step iii is to improve the training efficiency and accuracy. This can be done by adding more samples in subdomains where the rate of misclassification is higher. In mp-DO, these subdomains typically correspond to the boundaries between critical regions. Here, more points are added around those ones whose predicted labels are not correct or not with high enough confidence. The new points are generated randomly in an area with certain radius centred at those reference points, that approximately accounting for the region between the chosen reference points

Algorithm 4.1 MLP training procedure for mp-NCO-tracking.

Input: Uncertainty domain $\Theta \subset \mathbb{R}^{n_\theta}$; Confidence level $1-\alpha$ on a true classifier error rate ϵ; Number and size of hidden layers $H > 0$ in MLP.

Main loop:

i. Select an initial sample $\{\theta^{(1)}, \ldots, \theta^{(M)}\} \in \Theta$. Then, label each point $i = 1 \ldots M$ with $e_{k^{(i)}} \in \{0,1\}^K$, where $k^{(i)} \in \{1, \ldots, K\}$ is the (unique) matching switching structure, and $K \leq NS$ denotes the number of switching structures detected on the sample. Form the labeled training set $\mathcal{T} := \{(\theta^{(1)}, e_{k^{(1)}}), \ldots, (\theta^{(M)}, e_{k^{(M)}})\}$.

ii. Set-up an MLP with H hidden layers and K neurons in the output layer, and train it with the labeled set $\mathcal{T}$. If the evaluation of testing error rate is lower than the true classifier error rate with confidence $1-\alpha$, **stop**.

iii. Expand the training set $\mathcal{T}$ by adding extra points $i = M+1 \ldots$ based on the adaptive sampling strategy. Then, label these extra points with $e_{k^{(i)}} \in \{0,1\}^K$, and update the number K of detected switching structures accordingly. **Return** to Step ii.

Return: Trained MLP for mp-DO problem (3.1) and their neighbours. For our specific model type, the highest probability being under certain level indicates that the label is probably incorrect or near the region boundaries. Besides, additional randomly generated Sobol points are added into the new training set to to increasing the density and randomness from the global exploration aspect. There is furthermore no guarantee that all of the possible switching structures within the parameter set Θ have been uncovered, such that unexplored switching structures might exist in some very small area that totally enclosed by the recognized region and far from its boundaries. The possibility of any existence of unexplored regions may be decreased by applying a finer sampling density globally. Another way to guarantee the exploring entire critical region may be by solving the nonlinear problem (2.29). Furthermore, analysis based on the real physical systems might be considered to avoid such a situation happening from a system specific perspective.

3.3.4 On-line implementation of classifier

For the procedure of on-line control, the classifier is embedded into the mp-NCO-tracking controller for point location. A system parameter θ is taken as as an input to the trained MLP obtained by Algorithm 4.1, and the output is computed as a distribution for the potential critical regions $y(\theta) = [p_1, \ldots, p_k, \ldots, p_K]$; afterwards, the correct switching structure $\mathcal{S}^i$ is chosen based on value of p_k and its correspond switching times t_k are calculated as in step (II) in Section 3.3.1. In the case that the predicted class label with the highest score is not correct, such that the switching times obtained do not agree with the underlying switching structure, successive choices among the remaining class labels are made based on the probability distribution, with the order of choices based on p_k from high to low. Due to the computational accuracy restriction, the generalized prediction accuracy can not be guaranteed to be exactly 1.0. With the definition of critical regions having a physical meaning, all inaccuracies normally exist around the regions boundaries. In such a way, θ is mapped into Θ^i, which corresponds to the feedback laws $\mathcal{K}^{(i)}$, which is applied to the system until next update is available.

Furthermore, another advantage of the developed classifier lies in the fact it can perform parallel learning task separately from working as predictor. Also, in case of the parameter falls into an undetected region, where all the existing switching structures are incorrect, one may considering solving the QP for this specific parameter and recognize the corresponding switching structure. Additionally, this can be taken as a new sample example and the existing MLP model needs to be updated and further training, which should be done parallel with usage the model. In such a way, the accuracy and reliability should be improved with more training data feeding and validation, thus the generalization performance can be monotonically improving as time evolves.

3.4 Case studies

3.4.1 An FCC system

We consider a fluidized catalytic cracking (FCC) unit operated in partial combustion mode [85]. The objective is to steer the system to a given operating point, defined in terms of the mass fraction of coke on regenerated catalyst, C_{rc}, and the regenerator dense bed temperature, T_{rg}. The manipulated variables are the flow rate of air sent to the regenerator, F_{a}, and the catalyst flow rate, F_{s}. A linear input-output dynamic model is obtained via linearization and reduction of a first-principles nonlinear model around the equilibrium point $C^*_{\mathrm{rc}} = 5.207 \times 10^{-3}$, $T^*_{\mathrm{rg}} = 965.4\,\mathrm{K}$, $T^*_{\mathrm{ro}} = 776.9\,\mathrm{K}$, $T^*_{\mathrm{cy}} = 988.1\,\mathrm{K}$, and $T^*_{\mathrm{f}} = 400\,\mathrm{K}$, where T_{f} denotes the feed oil temperature. The control and state variables in this linear dynamic system are

$$x(t) := \begin{bmatrix} C_{\mathrm{rc}}(t) - C^*_{\mathrm{rc}} \\ T_{\mathrm{rg}}(t) - T^*_{\mathrm{rg}} \end{bmatrix}, \qquad u(t) := \begin{bmatrix} F_{\mathrm{s}}(t) - F^*_{\mathrm{s}} \\ F_{\mathrm{a}}(t) - F^*_{\mathrm{a}} \end{bmatrix}.$$

This optimization-based control problem is given by

$$\begin{aligned} \inf_{x,u}\ & \frac{1}{2}\int_0^{10} x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t)\ \mathrm{d}t + \frac{1}{2} x(T)^\mathsf{T} Q_{\mathrm{f}} x(T) \\ \text{s. t.}\ & \dot{x}(t) = F_x x(t) + F_u u(t) + F_\theta \theta_3 + f_0 \\ & g(x(t), u(t)) \leq 0 \\ & x(0) = (\theta_1, \theta_2). \end{aligned} \tag{3.21}$$

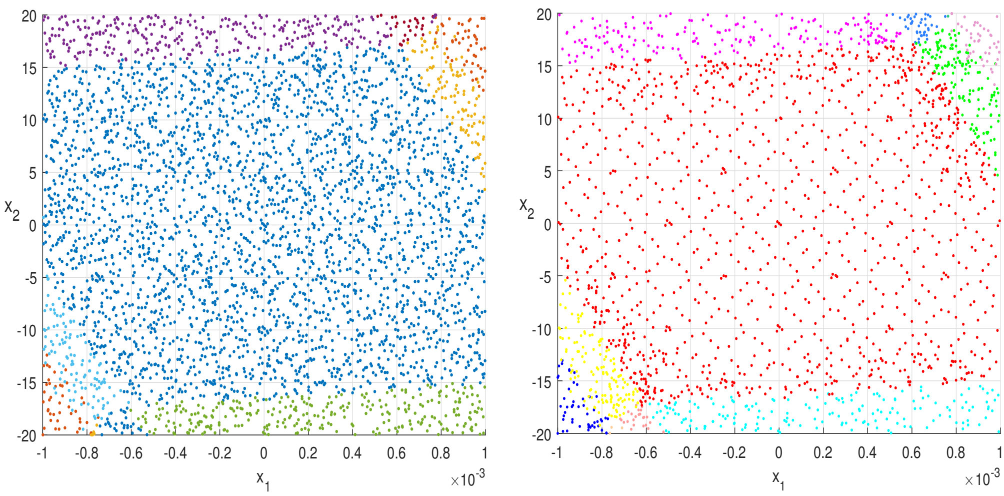
where detailed values for the matrices can be found in Section. 2.4.3. The parameters θ correspond to uncertainty in the initial values of C_{rc}, T_{rg} and T_{f}, with $\Theta = [-10^{-3}, 10^{-3}] \times [-20, 20] \times [-5, 5]$.

We apply Algorithm 4.1 to construct an MLP that describes the switching structures of the mp-DO (3.21). Here, all the data points are prepossessed to be in $[-1, 1]$ in each dimension. The selected MLP is a fully connected neural network $\mathcal{C}_\theta$, with one input layer, two hidden layers and one output layer layer. Each hidden layer contains twenty neurons with $\tanh(x)$ as the sigmoid activation functions in each neuron, and softmax function is used in the output layer. Upon termination of Algorithm 4.1, the trained MLP represents 11 critical regions, corresponding to the eleven neurons in the output layer of the neural network, and the details for different switching structures can be found in Section. 2.4.3.

We start with an initial Sobol set with 1000 sample points with an initial output vector of dimension 9 and testing accuracy 96.8%. Afterwards, two sampling strategies are applied for iteratively adding sample points in training set until the desired accuracy is reached. For comparison, the same testing set of 1000 points generated independently from training set is used for estimating the generalized true accuracy, such that two different classifiers aims to reach the same accuracy level. First, we add a fixed number of 200 randomly generated Sobol points across the entire domain in each iteration as a brute force exploration, and a final training set with 3800 points is obtained. The projection of resulting sample set on (θ_1, θ_2) is represented in Fig 3.2(a), using a different color for each switching structure. Such labelling is done by parameterizing the control trajectories as piecewise constant and discretizing the path constraints at interior points over 100 stages, and then solving the resulting QP problems using Gurobi. Next, the method of adaptively training is used based on the initial sample set, where more sample points in subdomains are generated and added to the training set. These subdomains, corresponding to areas with high nonlinearities, are chosen as the local regions within certain radius centred at selected training points, which are either misclassified or with a low value of the maximum element in output vector. Besides, additional 100 Sobol points are added to the training set to increase randomness and density for the space filling. The second training set with 1960 sample points in Θ is generated iteratively, which is shown in Fig 3.2(b). In this way, the total number of training set is smaller than the simply non-biased sampling method while still reaching the same accuracy level.

The testing accuracy is $\hat{\epsilon} = 99.9\%$ when taking into consideration the structure with highest probability in the softmax classifier for the test set with 1000 sample points, which is a Sobol set with uniform distribution. Thus, an estimation of generalized prediction accuracy $\epsilon = 99.7\%$ is accepted with confidence of 95% based on the prediction of highest probability. Furthermore, there is no misclassification in the testing set after taking into consideration the structure with second highest probability as well. This indicates that in the case of misclassification, a second solution is correct for the sample data in test set. Additional calculation of the testing accuracy of two independent testing sets with 500 and 2000 points are performed for illustration on accuracy test, with 1.0 for the former and 99.9% for the later, which corresponds to prediction accuracy $\epsilon = 99.75\%$ with confidence 95%. It can be shown that, the generalized performance is an estimation jointly from the testing accuracy and size of the testing size, and the 1.0 accuracy on one testing set does not indicate the exact true accuracy. Since the space is continuous, the global generalization performance can be only estimated with certain confidence statistically.

For the on-line control, the classifier takes the parameter θ as input and gives an output denoting the probability distribution $Y(\theta) = [y_1, \dots, y_{11}]$, where y_i corresponds to critical region CR_i. The switching structure $\mathcal{S}_i$ is then selected and the corresponding control laws Eq. 3.2 is applied until next feedback of the parameter.



(a) (b)

Figure 3.2: Sample data for training neural network

For the parameter value $\theta = (9 \cdot 10^{-4}, 14, 0)$, the structure with highest probability predicted by the softmax classifier is Θ^6 (99.99%). The corresponding switching structure is comprised of three arcs, with a boundary arc x_1 reaching its upper bound located in between an initial interior arc and a final arc. In turn, we can use the corresponding feedback laws $\mathcal{K}^6$ given in (3.2) to compute the switching times $t_1(\theta), t_2(\theta)$, and check that all of the corresponding primal and dual feasibility conditions are satisfied. The resulting optimal control trajectory is shown on Fig. 3.3.

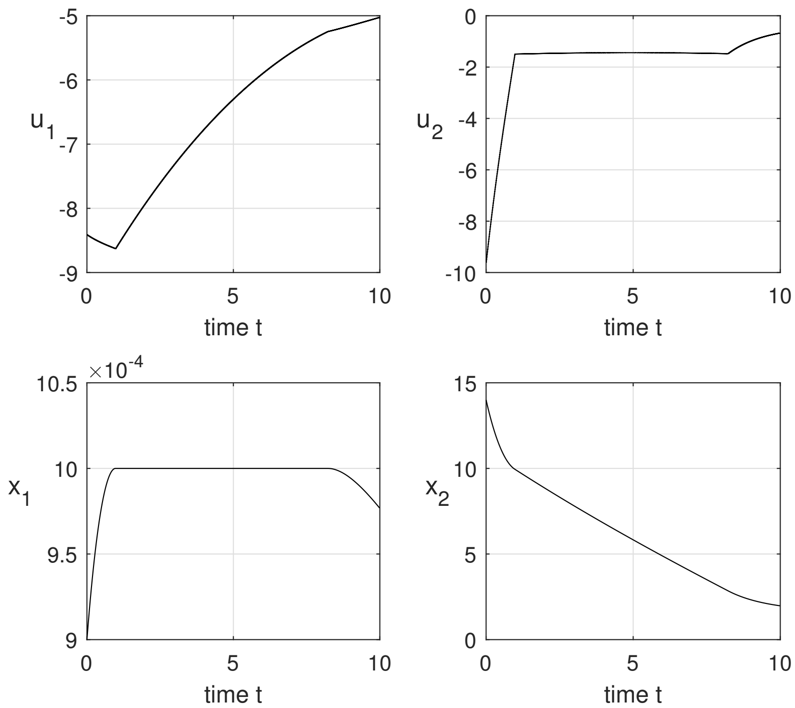


Figure 3.3: Trajectory for $\theta \in \mathsf{CR}_6$ by mp-DO solution

One of the misclassified samples in the training set is obtained for the parameter values $\theta = (9.2 \times 10^{-4}, 13, 5)$. The structure with highest probability that is predicted by the softmax classifier is again Θ^6 (55.53%). But in calculating the optimal switching times, we find that the switching between the second and final arcs should now occur at $t_2 \approx 10.012$, which is beyond the time horizon $T = 10$. Therefore, this point does not belong to Θ^6. The structure with second highest probability is Θ^5 (44.47%), which does not present the final interior arc, and turns out to be the correct one. The resulting optimal control trajectory is shown on Fig. 3.4.

3.4.2 A series of chemical reactors

Consider a system of two well mixed, non-isothermal continuous stirred tank reactors CSTRs operating in series [123], as shown in Fig. 3.5, where three parallel irreversible elementary exothermic reactions of the form $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} U$, and $A \xrightarrow{k_3} R$ take place. A is the reactant species, B is the desired product and U and R are undesired byproducts. The feed to the first reactor consists of flow rate F_0 with pure A, molar concentration C_{A0} and temperature T_0, and the output of the first reactor is fed to the second reactor. Another fresh feed to the second reactor consists of flow rate F_3 with pure A, molar

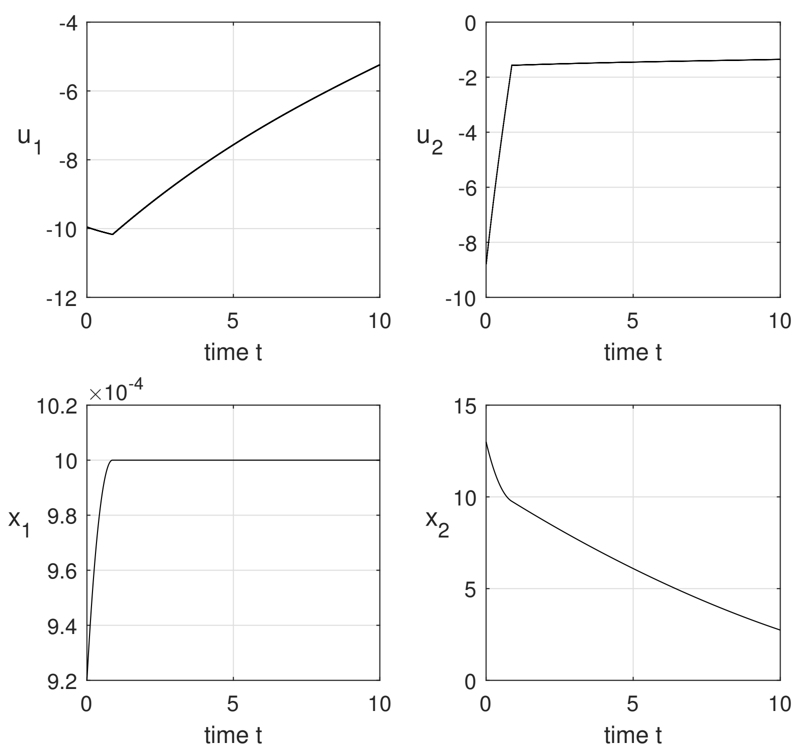


Figure 3.4: Trajectory for $\theta \in \mathsf{CR}_5$ by mp-DO solution

concentration C_{A03} and temperature T_{03}.

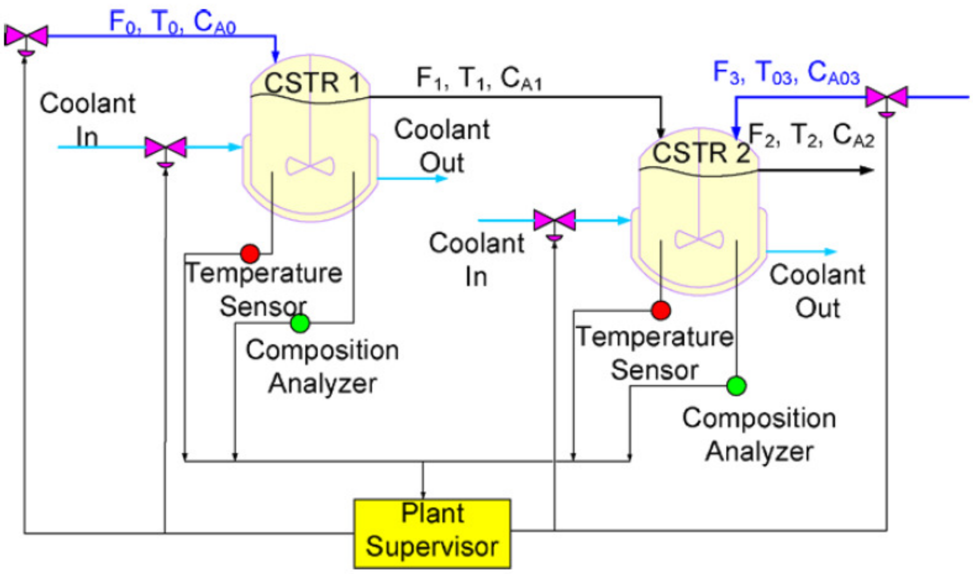


Figure 3.5: System of two CSTRs operating in series

The mathematical model of the process derived from material and energy balances is in Eq. (3.22), where T, C_A, Q and V denote the temperature of the reactor, the concentration of species A, the rate of heat input/removal from the reactor, and the volume of reactor, respectively, with subscript j denoting CSTR$_j$.

$$
\begin{aligned}
\frac{dC_{A1}}{dt} &= \frac{F_0}{V_1}(C_{A0} - C_{A1}) - \sum_{i=1}^{3} R_i(C_{A1}, T_1) \\
\frac{dT_1}{dt} &= \frac{F_0}{V_1}(T_0 - T_1) + \sum_{i=1}^{3} \frac{-\Delta H_i}{\rho c_p} R_i(C_{A1}, T_1) + \frac{Q_1}{\rho c_p V_1} \\
\frac{dC_{A2}}{dt} &= \frac{F_0}{V_2}(C_{A1} - C_{A2}) + \frac{F_3}{V_2}(C_{A03} - C_{A2}) + \sum_{i=1}^{3} R_i(C_{A2}, T_2) \\
\frac{dT_2}{dt} &= \frac{F_0}{V_2}(T_1 - T_2) + \frac{F_3}{V_2}(T_{03} - T_2) + \sum_{i=1}^{3} \frac{-\Delta H_i}{\rho c_p} R_i(C_{A2}, T_2) + \frac{Q_2}{\rho c_p V_2}
\end{aligned}
\tag{3.22}
$$

where $R_i(C_{Aj}, T_j) = k_{i0} exp(\frac{-E_i}{RT_j}) C_{Aj},\ j = 1, 2;\ \Delta H_i,\ k_i,\ E_i,\ i = 1, 2, 3,$ denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively; c_p and ρ denote the heat capacity and density of the fluid. Values of other constant parameters can be found in Table 3.1.

Table 3.1: Parameters for two CSTRs system

$F_0^* = 4.998$ m^3/hr	$F_3^* = 30.0$ m^3/hr
$V_1 = 1.0$ m^3	$V_2 = 3.0$ m^3
$T_0 = 300$ K	$T_{03} = 300$ K
$C_{A0}^* = 4.0$ kmol/m^3	$C_{A03}^* = 3.0$ kmol/m^3
$\Delta H_1 = -5.0 \times 10^4$ kJ/kmol	$\Delta H_2 = -5.2 \times 10^4$ kJ/kmol
$\Delta H_3 = -5.4 \times 10^4$ kJ/kmol	$k_{10} = 3.0 \times 10^6$hr^{-1}
$k_{20} = 3.0 \times 10^5$hr^{-1}	$k_{30} = 3.0 \times 10^5$hr^{-1}
$E_1 = 5.0 \times 10^4$ kJ/kmol	$E_2 = 7.53 \times 10^4$ kJ/kmol
$E_3 = 7.53 \times 10^4$ kJ/kmol	$R = 8.314$ kJ/kmolK
$\rho = 1000.0$ kg/m^3	$c_p = 0.231$ kJ/kgK

The control objective is to steer system to an unstable steady state $(C_{A1}^*, T_1^*, C_{A2}^*, T_2^*) = (3.59\ \text{kmol/m}^3, 388.57\ \text{K}, 2.55\ \text{kmol/m}^3, 429.24\ \text{K})$, with $Q_1 = 0$ and $Q_2 = 0$, while simultaneously maintaining reactant concentration and avoiding high temperatures in the presence of input actuations. Constraints for the input and state variables are $|F_0 - F_0^*| \leq 4.998$ m^3/hr, $|Q_1| \leq 1 \times 10^7$ kJ/hr, $|F_3 - F_3^*| \leq 30.0$ m^3/hr, $|Q_2| \leq 3 \times 10^7$ kJ/hr, $|C_{Aj} - C_{Aj}^*| \leq 0.1$ and $|T_j - T_j^*| \leq 20$ K, $j = 1, 2$. The manipulated inputs are chosen as F_0, Q_1, F_3, Q_2. A linear input-output dynamic model is obtained via linearization and reduction of a first-principles nonlinear model around the equilibrium point

$(C_{A1}^*, T_1^*, C_{A2}^*, T_2^*)$. The control and state variables in this linear dynamic system are

$$x(t) := \begin{bmatrix} C_{A1}(t) - C_{A1}^* \\ T_1(t) - T_1^* \\ C_{A2}(t) - C_{A2}^* \\ T_2(t) - T_2^* \end{bmatrix}, \qquad u(t) := \begin{bmatrix} F_0(t) - F_0^* \\ Q_1(t) \\ F_3(t) - F_3^* \\ Q_2(t) \end{bmatrix}.$$

The optimization-based control problem complies with the formulation in (3.1), with the terminal time $t_f = 5$. The non-zero matrices in the objective function and linear dynamic system of the optimal control problem are:

$$Q_{\mathrm{f}} = \begin{bmatrix} 2.71 & -6.64\times10^{-4} & 3.40\times10^{-1} & -2.44\times10^{-5} \\ 6.64\times10^{-4} & -1.23\times10^{-5} & 3.74\times10^{-4} & 4.47\times10^{-8} \\ -3.40\times10^{-1} & 3.74\times10^{-4} & 1.39\times10 & 1.15\times10^{-3} \\ -2.44\times10^{-5} & -4.47\times10^{-8} & 1.15\times10^{-3} & 4.28\times10^{-5} \end{bmatrix},$$

$$Q = \mathrm{diag}\left[1.00\times10^{2},\ \ 2.50\times10^{-3},\ \ 1.00\times10^{4},\ \ 1.56\times10^{-1}\right],$$

$$R = \mathrm{diag}\left[2.00\times10^{2},\ \ 1.00\times10^{-12},\ \ 5.00\times10^{-4},\ \ 2.50\times10^{-14}\right],$$

$$F_x = \begin{bmatrix} -5.57 & -8.15\times10^{-2} & 0 & 0 \\ 1.23\times10^{2} & 1.26\times10 & 0 & 0 \\ 1.67 & 0 & -1.41\times10 & -2.05\times10^{-1} \\ 0 & 1.67 & 5.34\times10^{2} & 3.27\times10 \end{bmatrix},$$

$$F_u = \begin{bmatrix} 4.09\times10^{-1} & 0 & 0 & 0 \\ -8.86\times10 & 4.33\times10^{-3} & 0 & 0 \\ 3.48\times10^{-1} & 0 & 1.51\times10^{-1} & 0 \\ -1.36\times10 & 0 & -4.31\times10 & 1.46\times10^{-3} \end{bmatrix}.$$

The path constraints $g(x(t), u(t))$ for the linearized system is

$$\begin{bmatrix} -4.998 \\ -1\times10^{7} \\ -30 \\ -3\times10^{7} \end{bmatrix} \le u(t) \le \begin{bmatrix} 4.998 \\ 1\times10^{7} \\ 30 \\ 3\times10^{7} \end{bmatrix}, \quad \begin{bmatrix} -0.1 \\ -20 \\ -0.1 \\ -20 \end{bmatrix} \le x(t) \le \begin{bmatrix} 0.1 \\ 20 \\ 0.1 \\ 20 \end{bmatrix}.$$

The parameters θ correspond to uncertainty in the initial values of state variables $x(0)$, with $\Theta = [-0.1, -20, -0.1, -20] \times [0.1, 20, 0.1, 20]$. The four dimensional space can not be visualized in general and the partition of the critical regions is represented by the classification model. A classifier is trained based on a neural network with an input layer containing 4 neurons, five hidden layers each containing 20 neurons with $\tanh(x) = (1 - e^{-2x})/(1 + e^{-2x})$ as activation function and an output layer containing 17 neurons. This means that there are 17 critical regions in total, and the direct output of the classifier for θ is a vector $Y(\theta) = [y_1, \ldots, y_{17}]$. We start with an initial Sobol set with 5000 sample points as the training set and iteratively adding sample points in training set. The testing data is generated independently from the training set by a Sobol set with 5000 uniformly distributed points, each attached with a label from 1 to 17. A final training set with 16294 points is obtained by 5 iterations. The labelling of the sample points is again by parameterizing the control trajectories as piecewise constant and discretizing the path constraints at interior points over 50 stages, and then solving the resulting QP problems using Gurobi. Next, the method of adaptively training is used based on the initial sample set, where more sample points in subdomains are generated and added to the training set.

Based on the the label corresponding to the highest probability, the testing accuracy 99.78% is obtained. When choosing a confidence of 95%, an estimation of the generalized accuracy of 99.66% is calculated based on binomial distribution. To illustrate the estimation of generalized accuracy, another independent testing of a Sobol set of 4000 shows an testing accuracy 99.80%, thus the corresponding estimation is with confidence of 95% is 99.68%, which also shows that the generalization performance is jointly estimated by the testing accuracy and size of the testing size. Among all the critical regions, the active constrains include saturations of the fourth input u_4 and the first state x_1 reaching its lower and upper bounds. Detailed description of the switching structures for each critical region is as follows:

- The critical region CR_1 corresponds to unconstrained optimal controls.
- The solutions in CR_2 and CR_3 are comprised of two arcs, a boundary arc where u_4 reaches its lower and upper bound, respectively, followed by an interior arc.
- The solutions in CR_4 and CR_5 are comprised of three arcs, with a boundary arc where x_1 reaches its upper and lower bound, respectively, located in between an

initial interior arc and a final one.

- The solutions in CR_6, CR_7, CR_8 and CR_9 combine the previous two cases in CR_2+CR_4 or CR_2+CR_5 or CR_3+CR_4 or CR_3+CR_5, and give rise to four arcs, starting with a boundary arc for u_4 (like CR_2 or CR_3), followed by an interior arc, a boundary arc for x_1 (like CR_4 or CR_5), and a final interior arc.

- The solutions in $\mathsf{CR}_{10} \sim \mathsf{CR}_{13}$ are comprised of four arcs, which are similar as for $\mathsf{CR}_6 \sim \mathsf{CR}_9$, except that the second arc corresponds two active constraints, u_4 and x_1 reaching their lower or upper bounds which are compatible with the first or third arc.

- The solutions in $\mathsf{CR}_{14} \sim \mathsf{CR}_{17}$ are comprised of four arcs, where the first three arcs contains a boundary arc for u_4 (like CR_2 or CR_3) with the second arc also contains a boundary arc for x_1 (like CR_4 or CR_5), followed by a final interior arc.

Take the initial state $\theta = [0.095,\ -19,\ 0.05,\ 10]$ as the input of classifier, an output vector is obtained with the fouth element being the largest value, which indicates that $\theta \in \mathsf{CR}_4$. And the input and state reference trajectories originating form θ based on the corresponding swithcing structure are shown in Fig. 3.7, where there are three arcs and the state variable x_1 reached its upper bound in the second arc.

Choose another parameter $\theta = [-0.095,\ 19,\ 0.09,\ -18]$, and the largest element of the output of classifier is y_8. The corresponding input and state reference trajectories originating form $\theta \in \mathsf{CR}_8$ are shown in Fig. 3.7, where the input variable u_4 reached its upper bound in the first arc and the state variable x_1 reached its lower bound in the third arc followed by the final interior arc.

The closed-loop simulations for the initial state $\theta = [-0.095,\ 19,\ 0.09,\ -18]$ by applying controller with $\Delta T = 1$ are shown in Fig. 3.8. When a parameter θ is given to the classifier, the corresponding swithcing structure is obtained based on the probability distribution for the critical regions. Here, the system input variables are calculated based on the linear system and closed-loop responses are obtained by simulations of the original non-linear dynamic system. The controller keeps the response following the optimal switching structures jumping from CR_8 to CR_5 to CR_1, and finally the system is stabilized at the desired steady state.

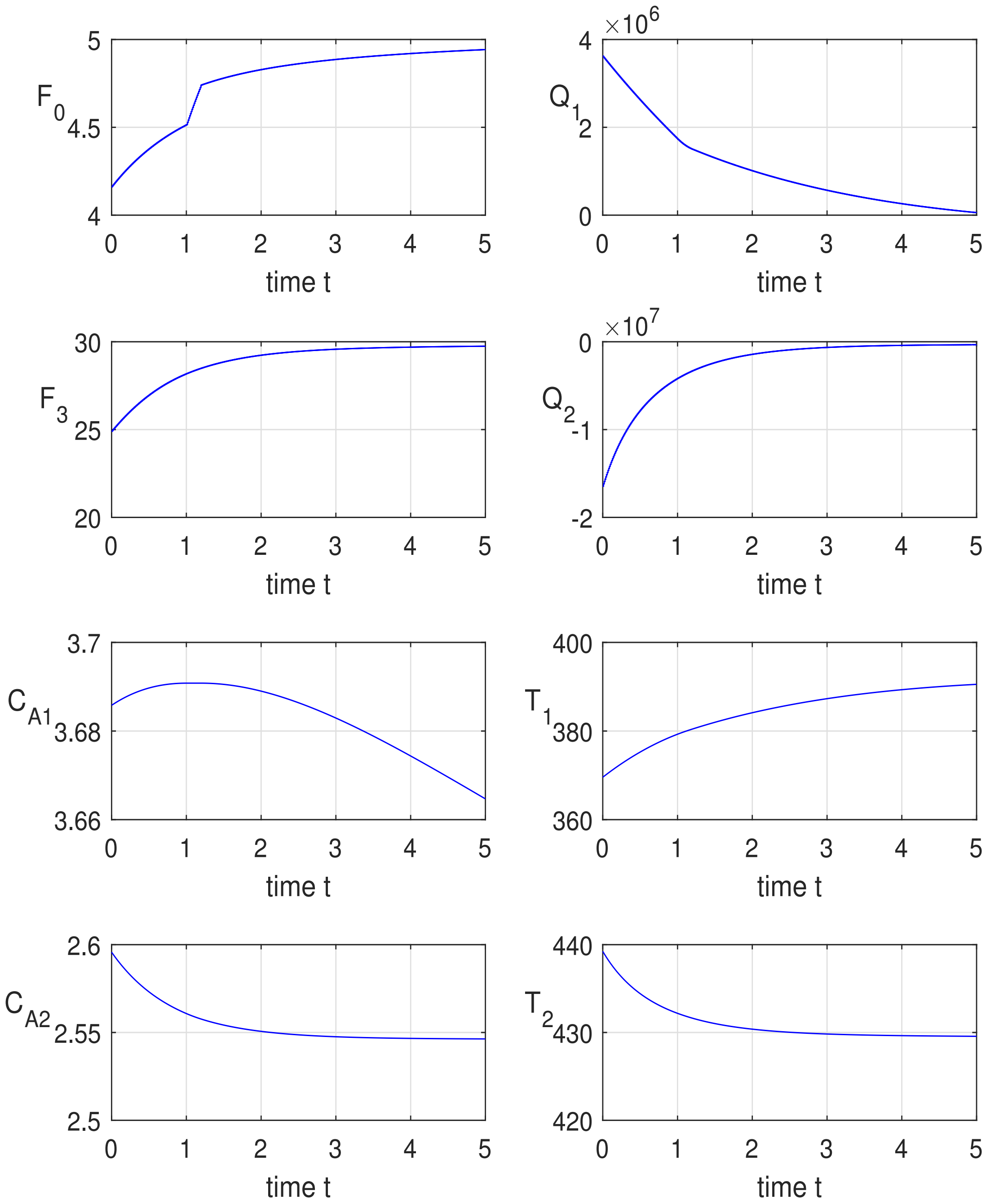


Figure 3.6: Optimal solution by mp-DO for CSTRs system in CR_4

3.5 Conclusion

This chapter presents an algorithm on training a classifier for recognition of critical regions to be integrated into the mp-NCO-tracking controller. The principal objective is the use of data classifiers based on deep learning to approximate the critical regions in continuous-time mp-DO problems, and subsequently search for a critical region during on-line execution. A neural network based model is selected for the classification problem due to its representational power as universal approximator for continuous functions. The

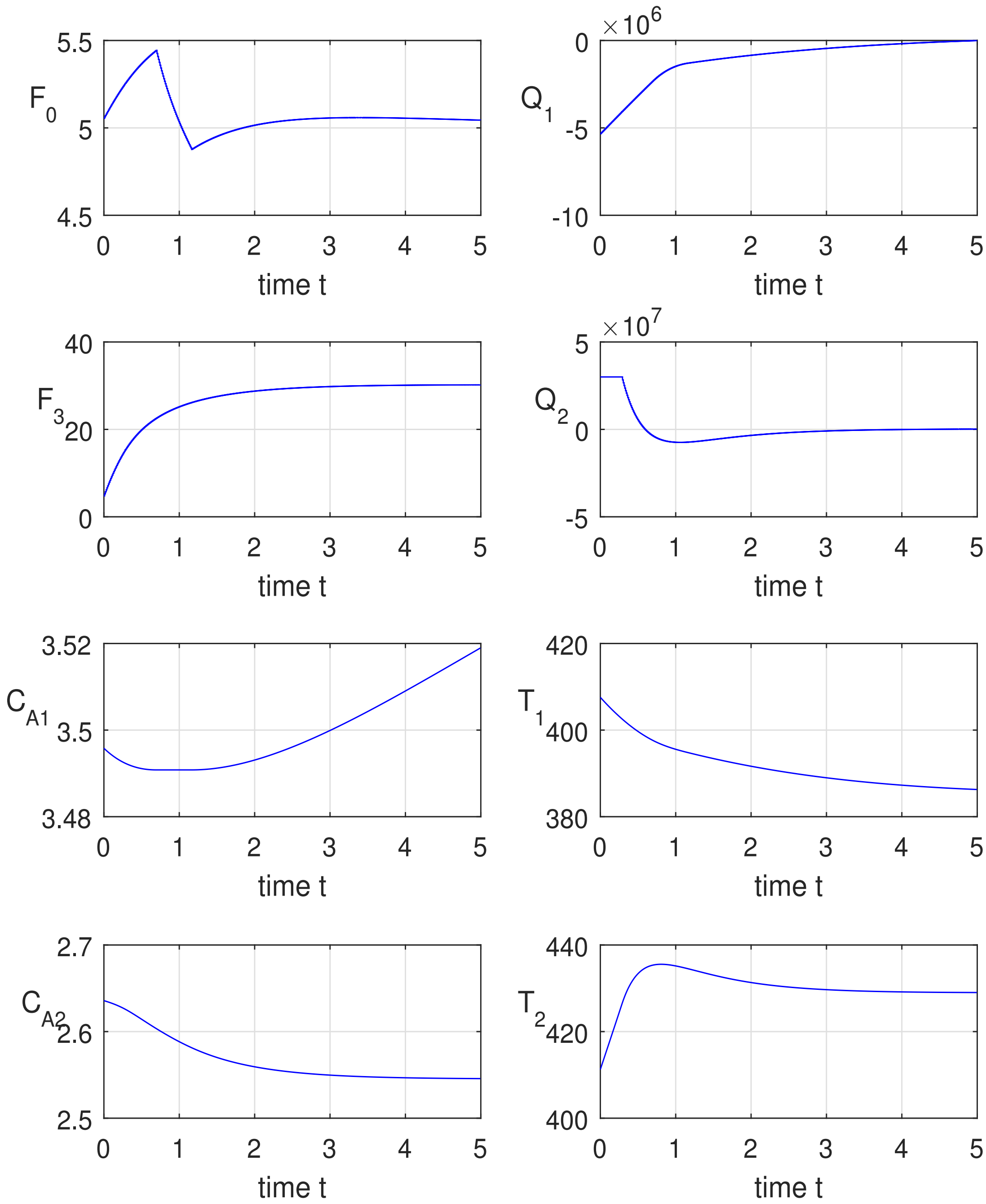


Figure 3.7: Optimal solution by mp-DO for CSTRs system in CR_8

detailed layer-wise architecture and mathematical formulation of the model are presented, including the definition of the score function for computing class scores and the loss function. The objective for training the model is to minimize the loss function defined by the cross entropy between the prediction and the ground truth labels. In the output layer of the network, softmax classifier is used to generate a normalized probability distribution on all the critical regions, given a parameter of the control problem as input. And it also has the potential ability to handle the case of misclassification. General steps for setting up the model and training based on optimization are discussed, and evaluation of

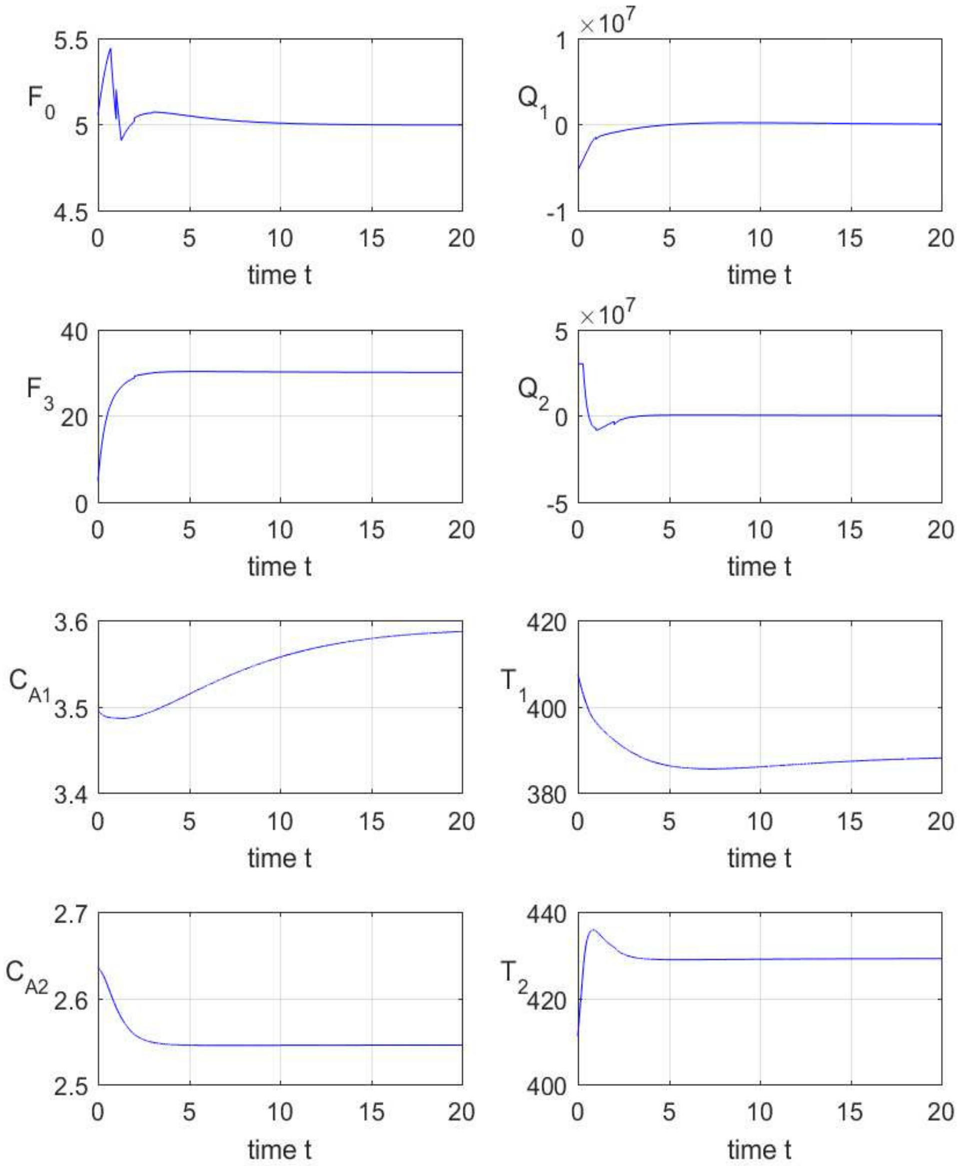


Figure 3.8: Closed-loop response of the mp-NCO-tracking controllers for CSTRs system.

the generalized accuracy on prediction of the classifier is performed based on uniformly distributed random data.

The algorithm for the implementation of the mp-NCO-tracking controller is presented in a step-wise procedure. This calls for a training set with labelled points, a neural network based classification model with a layer-wise structure and an optimizer to compute the value of parameters for the neural network, namely, the weights and bias that characterized the classification model. After the training, sample data can be completely discarded, and the classifier is used for locating the measurement of the systems parameter into critical regions for on-line control. Estimation of the true accuracy of the network is computed based on a binomial distribution, and the situation that points around region boundaries are misclassified is also considered by taking the highest few candidate classes obtained directly from the output layer. Finally, these developments are illustrated by case studies of an FCC system and a system of two CSTRs operated in series, where classifiers are trained for reaching a certain accuracy on the prediction and applied in the closed-loop control of the dynamic systems.

Chapter 4

Robust multi-parametric NCO-tracking control

4.1 Introduction

This chapter extends the mp-NCO-tracking approach to enable robust multi-parametric controllers for continuous-time linear dynamic systems. One limitation within the framework of mp-NCO-tracking is that the controllers are based on a certainty-equivalence principle, whereby the future of the system is optimized as if neither external disturbances nor model mismatch were present, despite the fact that such disturbances and mismatch are the reason why feedback is needed in the first place. This situation is similar to classical MPC [32, 124] and is a clear call for the development of robust mp-NCO-tracking controllers.

Popular approaches to designing robust MPC controllers for linear systems rely on the construction of ancillary feedback laws using linear matrix inequalities (LMI) and min-max optimization [125–127]; the propagation of robust forward invariant tubes enclosing the state trajectories for all possible disturbance realizations [128–133]; and the reformulation of MPC as a multi-stage optimization problem using dynamic programming and subsequent robustification of the subproblems [134]. The approach developed in this chapter for robustifying mp-NCO-tracking controllers is inspired by tube-based MPC. It involves backing-off the path and terminal state constraints based on a worst-case uncertainty propagation determined using either interval analysis or ellipsoidal calculus.

An important advantage of this approach is that it retains the same complexity as with nominal mp-NCO-tacking controllers. Preliminary results have been presented in [135], which are extended herein by introducing ancialiry feedback laws in order to reduce the conservatism.

The focus herein is on constructing robust mp-NCO-tracking controllers, which can guarantee feasibility in the worst-case scenario for the time-varying uncertainty $w(t)$. The main objective is to develop robust formulations that are amenable to numerical solution within the same computational effort as the nominal mp-NCO-tracking controllers in [96], i.e. with $F_w = 0$. This way, the developments in Chapter 2 and chapter 3 are ready to be applied in the robust control. The system state is decomposed into two parts, a nominal reference component and a disturbed component. The dynamics of disturbed component evolves separately from the nominal dynamics, and convex enclosure of the uncertain dynamics is computed off-line for the worst-case scenario based on interval or ellipsoidal reachability tubes centred at the nominal reference trajectory. This way, the corresponding back-offs on constraints based on these reachability tubes can be computed prior to solving the mp-DO problem, thus enabling the direct application of the controller design procedure. Finally, the robust counterpart of the control problem with unknown disturbances is formulated by backing-off the constraints. To reduce conservatism and stabilize the reachability tubes, additional linear state feedback input with respect is added to the nominal input for the reference trajectory. These developments are illustrated with the case studies of a fluidized catalytic cracking (FCC) unit operated in partial combustion mode and a series of chemical reactors.

The rest of the chapter is organized as follows: Section 4.2 lists a couple of notations which are essential for the formulation of robust counterpart of the control problem; Section 4.3 presents the formulation of the robust control problem studied in this chapter; Section 4.4 develops robust mp-NCO-tracking controllers based on a robust-counterpart formulation of the mp-DO problem, considering both additive and multiplicative time-varying uncertainty, and training classifiers for critical regions; results for the numerical case studies of a fluidized catalytic cracking (FCC) unit and a series of chemical reactors are presented in Section 4.5; finally, Section 4.6 concludes the chapter.

4.2 Notations

The sets of compact and convex subsets of $\mathbb{R}^n$ are denoted by $\mathbb{K}^n$ and $\mathbb{K}^n_{\mathrm{C}}$. The set of n-dimensional symmetric positive semi-definite (resp. positive definite) matrices is denoted by $\mathbb{S}^n_+$ (resp. $\mathbb{S}^n_{++}$).

An interval vector is denoted by $\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]$, and its radius and midpoint are defined as:

$$\begin{aligned}
\operatorname{mid}\left(\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]\right) &:= \frac{\left(y^{\mathrm{U}} + y^{\mathrm{L}}\right)}{2}\,, \\
\operatorname{rad}\left(\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]\right) &:= \frac{\left(y^{\mathrm{U}} - y^{\mathrm{L}}\right)}{2}\,.
\end{aligned}$$

An ellipsoid with center $q \in \mathbb{R}^n$ and shape matrix $Q \in \mathbb{S}^n_+$ is given by

$$\mathcal{E}(q,Q) := \left\{ q + Q^{\frac{1}{2}} v \;\middle|\; v \in \mathbb{R}^n, v^{\mathsf{T}} v \leq 1 \right\}\,.$$

The support function $V[Z] : \mathbb{R}^n \to \mathbb{R}$ of a set $Z \in \mathbb{K}^n$ is

$$\forall c \in \mathbb{R}^n, \quad V[K](c) := \max_z \{ c^{\mathsf{T}} z \,|\, z \in Z \}\,.$$

In particular, the support functions of the interval $\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]$ and the ellipsoid $\mathcal{E}(q,Q)$ are defined for each $c \in \mathbb{R}^n$ as

$$\begin{aligned}
&V[\mathcal{E}(q,Q)](c) = c^{\mathsf{T}} q + \sqrt{c^{\mathsf{T}} Q c} \\
&V[y^{\mathrm{L}}, y^{\mathrm{U}}](c) = c^{\mathsf{T}} \operatorname{mid}\left(\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]\right) + \operatorname{abs}(c)^{\mathsf{T}} \operatorname{rad}\left(\left[y^{\mathrm{L}}, y^{\mathrm{U}}\right]\right)
\end{aligned}$$

with $\operatorname{abs}(c) := (|c_1|, \ldots, |c_n|)^{\mathsf{T}}$.

The i-th row (resp. column) of a matrix $A \in \mathbb{R}^{n \times n}$ is denoted by $A_{(i,\cdot)}$ (resp. $A_{(\cdot,i)}$). The $m \times n$-matrix with zeros in all its entries is denoted by $0_{m \times n}$ or 0_n if $m = n$. The n-dimensional identity matrix is denoted by I_n.

Let $I \subset \mathbb{R}$, $\mathbb{L}_2^{n_u}(I)$ denotes the set of $\mathbb{L}_2$-integrable vector-valued functions of dimension n_u on I, and $\mathbb{W}_{1,2}^{n_x}(I)$, the set of n_x-dimensional weakly differentiable functions on I, whose weak derivative is $\mathbb{L}_2$-integrable.

Let $D_x^i f$ denotes the i-th partial derivative of a function f with respect to x and $f^{(j)}$, the j-th order derivative of with respect to t.

A path constraint $g_i = G_{x,i}x + G_{u,i}u + G_{\theta,i}\theta + g_{0,i}$ is said to be of order (or degree) $\sigma_i \geq 0$ if $D_u g_i^{(j)} \equiv 0$ for $j = 0 \dots \sigma_i - 1$ and $D_u g_i^{(\sigma_i)} \neq 0$. For simplicity, we use the notation as in Chapter 2

$$g_i^{(\sigma)}(x,u) := G_{x,i}^{(\sigma)}x + G_{u,i}^{(\sigma)}x + G_{\theta,i}^{(\sigma)}\theta + g_{0,i}^{(\sigma)}\,,$$

We also make use of the notation for $g = [g_1, \dots, g_{n_g}]^\mathsf{T}$, such that

$$\begin{aligned}
&g^{(\sigma)}(x,u) := G_x^{(\sigma)}x + G_u^{(\sigma)}u(t) + G_\theta^{(\sigma)}\theta + g_0^{(\sigma)}\,,\\
&\text{with}\quad G_x^{(\sigma)} := \begin{bmatrix} G_{x,1}^{(\sigma_1)} \\ \vdots \\ G_{x,n_g}^{(\sigma_{n_g})} \end{bmatrix}, \quad G_u^{(\sigma)} := \begin{bmatrix} G_{u,1}^{(\sigma_1)} \\ \vdots \\ G_{u,n_g}^{(\sigma_{n_g})} \end{bmatrix}, \quad G_\theta^{(\sigma)} := \begin{bmatrix} G_{\theta,1}^{(\sigma_1)} \\ \vdots \\ G_{\theta,n_g}^{(\sigma_{n_g})} \end{bmatrix}, \quad g_0^{(\sigma)} := \begin{bmatrix} g_{0,1}^{(\sigma_1)} \\ \vdots \\ g_{0,n_g}^{(\sigma_{n_g})} \end{bmatrix}.
\end{aligned}$$

An over-bar is used to indicate subsets of constraints that are active, such as $\bar{g}(x(t),u(t)) = \bar{G}_x x(t) + \bar{G}_u u(t) + \bar{G}_\theta\theta + \bar{g}_0 \leq 0$ and the Lagrangian multiplier $\bar{\mu}(t)$.

4.3 Problem Formulation

We consider constrained linear-quadratic optimal control problems under uncertainty, in the form:

$$\begin{aligned}
\inf_{x,u}\;\; & \frac{1}{2}\int_0^T x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t)\,\mathrm{d}t + \frac{1}{2}x(T)^\mathsf{T} Q_\mathrm{f} x(T)\\
\text{s.t.}\;\; & \dot{x}(t) = F_x x(t) + F_u u(t) + F_\theta\theta + F_w w(t) + f_0\\
& x(0) = B_0\theta + b_0\\
& G_x x(t) + G_u u(t) + G_\theta\theta + g_0 \leq 0\\
& H_x x(T) + H_\theta\theta + h_0 \leq 0\,.
\end{aligned} \tag{4.1}$$

Besides the standard notations defined previously, $x \in \mathbb{W}_{1,2}^{n_x}$ denote the state response; $u \in \mathcal{U} := \{u \in \mathbb{L}_2^{n_u} \,|\, \forall t \in [0,T], u(t) \in U\}$ with $U \in \mathbb{K}_\mathrm{C}^{n_u}$, the control input; $\theta \in \mathbb{K}_\mathrm{C}^{n_\theta}$, the problem parameters; $w \in \mathcal{W} := \{w \in \mathbb{L}_2^{n_w} \,|\, \forall t \in [0,T], w(t) \in W\}$ with $W \in \mathbb{K}_\mathrm{C}^{n_w}$, the time-varying uncertainty; and $Q_{\mathrm{f}} \succeq 0$, $Q \succeq 0$, and $R \succ 0$ are given weighting matrices. We shall often use the short-hand notations $f(x(t), u(t), \theta)$, $g(x(t), u(t), \theta)$, $h(x(T), \theta)$ and $J(x, u)$ to denote the ODE right-hand side, path constraint, terminal constraint and cost functions, respectively.

The focus of the chapter is on constructing robust mp-NCO-tracking controllers, in order to guarantee feasibility in the worst-case scenario for the time-varying uncertainty $w(t)$. The objective is to develop robust formulations that are amenable to numerical solution with the same computational effort as the nominal mp-NCO-tracking controllers in [96], i.e. with $F_w = 0$.

4.4 Robust mp-NCO-Tracking Controllers

This section develops a robustification of the mp-NCO-tracking methodology by [96], whereby worst-case uncertainty propagation [80] is applied to back-off the terminal and path constrained in order for the feedback controller to guarantee feasibility under all possible uncertainty scenarios. Furthermore, the conservatism of the open-loop solutions can be reduced by adding an additional state feedback control function $v : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ to the open-loop solution, which in the case of a linear feedback gain K leads to an ancillary state feedback law in the form

$$u(t, x) := \begin{cases} \hat{u}(t) & \text{if } t \in [0, t_{\mathrm{s}}) \\ \hat{u}(t) + K\left[x(t) - \hat{x}(t)\right] & \text{if } t \in [t_{\mathrm{s}}, T] \end{cases} \tag{4.2}$$

where $\hat{x}(t) \in \mathbb{R}^{n_x}$ and $\hat{u}(t) \in \mathbb{R}^{n_u}$ are the state and control values of the nominal system that generates the reference trajectory at time t; and $K \in \mathbb{R}^{n_u \times n_x}$ is a linear feedback gain matrix; and $t_{\mathrm{s}} > 0$ is the sampling period. In particular, this formulation distinguishes the first sampling period, during which the control actions are essentially in open loop. For the purpose of reducing conservatism and stability, the feedback gain K is chosen so as for the reachability tube for the disturbed component to be small enough and be controlled invariant at terminal time.

For simplicity, we shall use the following notations

$$\tilde{F}_x := F_x + F_u K$$

$$\tilde{G}_x := G_x + G_u K$$

throughout the developments in this chapter, and both matrices are directly affected by the value of the gain matrix K.

Regarding the uncertain system with disturbances, by substitution of $u(t)$ with (4.2), the closed-loop dynamics of the uncertain system in (4.1) are thus described by

$$\begin{aligned}
\dot{x}(t) &= F_x x(t) + F_u u(t,x) + F_\theta \theta + F_w w(t) + f_0 \\
&= \begin{cases} F_x x(t) + F_u \hat{u}(t) + F_\theta \theta + F_w w(t) + f_0 & \text{if } t \in [0, t_\mathrm{s}) \\ \tilde{F}_x x(t) - F_u K \hat{x}(t) + F_u \hat{u}(t) + F_\theta \theta + F_w w(t) + f_0 & \text{if } t \in [t_\mathrm{s}, T] \end{cases}
\end{aligned}$$

For a given uncertainty reference $\hat{w}(t) \in W$, we may then split the state and the disturbance into nominal and perturbed components as

$$x(t) = \hat{x}(t) + d_x(t)\,, \tag{4.3}$$

$$w(t) = \hat{w}(t) + d_w(t)\,, \tag{4.4}$$

so that the dynamics of the state reference $\hat{x}$ are given by

$$\begin{aligned}
&\dot{\hat{x}}(t) = F_x \hat{x}(t) + F_u \hat{u}(t) + F_\theta \theta + F_w \hat{w} + f_0 \\
&\text{with} \quad \hat{x}(0) = B_0 \theta + b_0\,.
\end{aligned} \tag{4.5}$$

and the dynamics of the perturbation d_x are given by

$$\begin{aligned}
&\dot{d}_x(t) = \begin{cases} F_x d_x(t) + F_w d_w(t) & \text{if } t \in [0, t_\mathrm{s}) \\ \tilde{F}_x d_x(t) + F_w d_w(t) & \text{if } t \in [t_\mathrm{s}, T] \end{cases} \\
&\text{with} \quad d_x(0) = 0\,.
\end{aligned} \tag{4.6}$$

Observe that the initial value problem (4.6) is independent of the nominal control $\hat{u}$ and the parameter θ. By linearity, a unique solution to the initial value problem (4.6) exists for all $w \in \mathcal{W}$, denoted by $\delta(\cdot, w)$ subsequently, and the reachability tube $\Delta : [0,T] \to \mathbb{K}^{n_x}$ describing the solution set of (4.6) for all possible realizations of the time-varying uncertainty w is given by

$$\Delta(t) := \{d \in \mathbb{R}^{n_x} \mid \forall t \in [0, T],\ \exists w \in \mathcal{W}:\ d = \delta(t, w)\}\ . \tag{4.7}$$

An illustration for the enclosure of reachability tubes is shown in Fig 4.1.

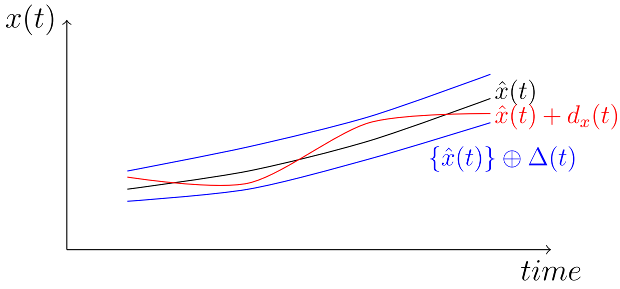


Figure 4.1: Enclosure of reachability tubes

The path constraints $g(x(t), u(t))$ can be formulated as

$$\begin{aligned} g(x(t), u(t, x)) &= G_x x(t) + G_u u(t, x) + G_\theta \theta + g_0 \\ &= \begin{cases} G_x x(t) + G_u \hat{u}(t) + G_\theta \theta + g_0 & \text{if } t \in [0, t_\mathrm{s}) \\ \tilde{G}_x x(t) + G_u \hat{u}(t) - G_u K \hat{x}(t) + G_\theta \theta + g_0 & \text{if } t \in [t_\mathrm{s}, T] \end{cases} \end{aligned}$$

Finally, the nominal dynamics, path and terminal constraints can be re-written as

$$\begin{aligned} \dot{\hat{x}}(t) &= F_x \hat{x}(t) + F_u \hat{u}(t) + F_\theta \theta + F_w \hat{w} + f_0 \\ g(x(t), u(t, x)) &= \begin{cases} G_x \hat{x}(t) + G_u \hat{u}(t) + G_\theta \theta + g_0 + G_x d_x(t) & \text{if } t \in [0, t_\mathrm{s}) \\ G_x \hat{x}(t) + G_u \hat{u}(t) + G_\theta \theta + g_0 + \tilde{G}_x d_x(t) & \text{if } t \in [t_\mathrm{s}, T]. \end{cases} \\ h(x(T)) &= H_x \hat{x}(T) + H_\theta \theta + h_0 + H_x d_x(t) \end{aligned} \tag{4.8}$$

The additional terms $G_x d_x(t)$, $\tilde{G}_x d_x(t)$ and $H_x d_x(t)$ in the above equations will potentially change the constraints compared to the nominal case, thus changing the types and shapes of the critical regions compared to the nominal case, since they are defined based on sequence of active constraints. Due to the constraints not remaining the same, following which the NCO conditions are different as well, nominal feedback laws will potentially change in each region based on the new constraints. For example, pure inputs constraint now become state-input constraints, and the admissible region shrinks as well.

It can be seen that the feedback gain K has a direct effect in perturbation dynamics and constraints through (4.6) and (4.8), such that the robust solution of the nominal component will rely on the choice of K. A smaller gain might result in more conservative solutions, while a larger gain might keep the perturbation relatively smaller but with the price of more aggressive control actions. Thus a trade-off exists for the closed-loop performance with state feedback inputs, and the overall performance also rely on the postpone realization of the disturbances, which may lead to smaller deviations than in the worst-case scenarios.

Provided that a compact reachable-tube enclosure $\bar{\Delta}(t) \supseteq \Delta(t)$ is available on $[0, T]$, a conservative robust counterpart to the problem (4.1) that minimizes the nominal cost may be formulated as [136]

$$\begin{aligned}
\inf_{\hat{x},\hat{u}} \quad & \frac{1}{2}\int_0^T \hat{x}(t)^\mathsf{T} Q \hat{x}(t) + \hat{u}(t)^\mathsf{T} R \hat{u}(t)\ \mathrm{d}t + \frac{1}{2}\hat{x}(T)^\mathsf{T} Q_\mathrm{f} \hat{x}(T) \\
\text{s.t.} \quad & \dot{\hat{x}}(t) = F_x \hat{x}(t) + F_u \hat{u}(t) + F_\theta \theta + F_w \hat{w} + f_0 \\
& \hat{x}(0) = B_0 \theta + b_0 \\
& \max_{\delta \in \bar{\Delta}(t)} G_x \hat{x}(t) + G_u \hat{u}(t) + G_\theta \theta + g_0 \leq -\delta_G \\
& \max_{\delta \in \bar{\Delta}(T)} H_x \hat{x}(T) + H_\theta \theta + h_0 \leq -\delta_H \,,
\end{aligned} \tag{4.9}$$

with

$$\delta_G := \begin{cases} \max\limits_{\delta \in \bar{\Delta}(t)} G_x \delta & \text{if } t \in [0, t_\mathrm{s}) \\ \max\limits_{\delta \in \bar{\Delta}(t)} \tilde{G}_x \delta & \text{if } t \in [t_\mathrm{s}, T] \end{cases} \quad \text{and} \quad \delta_H := \max_{\delta \in \bar{\Delta}(T)} H_x \delta \,.$$

An inherent advantage of this formulation is that the developments in [96] apply readily to the problem (4.9) in order to devise a robust mp-NCO-tracking controller. It is nonetheless clear that the back-off terms on the right-hand side of the robustified constraints will modify the boundary of the critical regions, and possibly the number and type of the critical regions too. Such changes in activation level of the constraints will furthermore propagate through the optimality conditions of the robustified mp-DO problem, thereby resulting in different feedback laws in those critical regions where path or terminal constraints are active. Provided the nominal controlled system being asymptotically stable at the origin and the enclosure of the reachable set being robustly controlled invariant, the closed-loop system is ultimately bounded in $\bar{\Delta}(T)$, which serves as the 'origin' of the uncertain controlled system [137].

In addition to a direct effect on the back-off terms of the robustified path constraints in (4.9), the gain matrix K will also have an indirect effect on the spread of the reachability tube enclosure $\bar{\Delta}$. In general, any set-valued function $\bar{\Delta} : [0,T] \to \mathbb{K}_{\mathrm{C}}^{n_x}$ satisfying the following generalized differential inequality for every $c \in \mathbb{R}^{n_x}$,

$$\text{a.e. } t \in [0, t_{\mathrm{s}}), \quad \dot{V}[\bar{\Delta}(t)](c) \geq \max_{d,w} \left\{ c^{\mathsf{T}} \left(F_x d + F_w w \right) \;\middle|\; \begin{array}{l} c^{\mathsf{T}} d = V[\bar{\Delta}(t)](c), \\ d \in \bar{\Delta}(t), \;\; w \in W \end{array} \right\}$$

$$\text{a.e. } t \in [t_{\mathrm{s}}, T], \quad \dot{V}[\bar{\Delta}(t)](c) \geq \max_{d,w} \left\{ c^{\mathsf{T}} \left(\tilde{F}_x d + F_w w \right) \;\middle|\; \begin{array}{l} c^{\mathsf{T}} d = V[\bar{\Delta}(t)](c), \\ d \in \bar{\Delta}(t), \;\; w \in W \end{array} \right\}$$

with $V[\bar{\Delta}(0)](c) \geq 0$, yields a valid enclosure of the reachable tube (4.7); see, e.g., [138]. The special cases of tractable interval and ellipsoidal enclosures are treated below, after which we discuss an approach to computing the gain matrix K.

4.4.1 Case of Interval Tube Enclosures

An interval enclosure $\overline{\Delta}(t) := \left[d_x^{\mathrm{L}}(t), d_x^{\mathrm{U}}(t) \right] \supseteq \Delta(t)$ can be precomputed via the following system of auxiliary ODEs:

$$\begin{aligned}
&\text{a.e. } t \in [0, t_{\mathrm{s}}), \quad \forall i \in 1, \ldots, n_x, \\
&\quad \dot{d}_{x_i}^{\mathrm{L}}(t) = \min_{\delta_x} \left\{ F_{x(i,\cdot)} \delta_x \;\middle|\; \begin{array}{l} \delta_{x_i} = d_{x_i}^{\mathrm{L}}(t) \\ \delta_x \in \bar{\Delta}(t) \end{array} \right\} + \min_{\delta_w} \left\{ F_{w(i,\cdot)} \delta_w \;\middle|\; \delta_w \in W \ominus \{\hat{w}(t)\} \right\}, \\
&\quad \dot{d}_{x_i}^{\mathrm{U}}(t) = \max_{\delta_x} \left\{ F_{x(i,\cdot)} \delta_x \;\middle|\; \begin{array}{l} \delta_{x_i} = d_{x_i}^{\mathrm{U}}(t) \\ \delta_x \in \bar{\Delta}(t) \end{array} \right\} + \max_{\delta_w} \left\{ F_{w(i,\cdot)} \delta_w \;\middle|\; \delta_w \in W \ominus \{\hat{w}(t)\} \right\}, \\
&\text{a.e. } t \in [t_{\mathrm{s}}, T], \quad \forall i \in 1, \ldots, n_x, \\
&\quad \dot{d}_{x_i}^{\mathrm{L}}(t) = \min_{\delta_x} \left\{ \tilde{F}_{x(i,\cdot)} \delta_x \;\middle|\; \begin{array}{l} \delta_{x_i} = d_{x_i}^{\mathrm{L}}(t) \\ \delta_x \in \bar{\Delta}(t) \end{array} \right\} + \min_{\delta_w} \left\{ F_{w(i,\cdot)} \delta_w \;\middle|\; \delta_w \in W \ominus \{\hat{w}(t)\} \right\}, \\
&\quad \dot{d}_{x_i}^{\mathrm{U}}(t) = \max_{\delta_x} \left\{ \tilde{F}_{x(i,\cdot)} \delta_x \;\middle|\; \begin{array}{l} \delta_{x_i} = d_{x_i}^{\mathrm{U}}(t) \\ \delta_x \in \bar{\Delta}(t) \end{array} \right\} + \max_{\delta_w} \left\{ F_{w(i,\cdot)} \delta_w \;\middle|\; \delta_w \in W \ominus \{\hat{w}(t)\} \right\}, \\
&\quad \text{with} \quad d_{x_i}^{\mathrm{L}}(0) = d_{x_i}^{\mathrm{U}}(0) = 0\,.
\end{aligned} \tag{4.10}$$

It follows that the robust counterpart problem (4.1) can be rewritten as

$$
\begin{aligned}
\inf_{\hat{x},\hat{u}} \quad & \frac{1}{2}\int_0^T \hat{x}(t)^\mathsf{T} Q\hat{x}(t) + \hat{u}(t)^\mathsf{T} R\hat{u}(t)\ \mathrm{d}t + \frac{1}{2}\hat{x}(T)^\mathsf{T} Q_\mathrm{f}\hat{x}(T) \\
\text{s. t.} \quad & \dot{\hat{x}}(t) = F_x\hat{x}(t) + F_u\hat{u}(t) + F_\theta\theta + F_w\hat{w} + f_0 \\
& \hat{x}(0) = B_0\theta + b_0 \\
& G_x\hat{x}(t) + G_u\hat{u}(t) + G_\theta\theta + g_0 \le -\delta_G(t) \\
& H_x\hat{x}(T) + H_\theta\theta + h_0 \le -\delta_H\,,
\end{aligned}
\tag{4.11}
$$

with the back-offs $\delta_G(t)$, $0 \le t \le T$, and δ_H taken as

$$
\delta_{G_i}(t) := \begin{cases} \operatorname{abs}\left(G_{x(i,\cdot)}\right)\operatorname{rad}\left(\overline{\Delta}(t)\right) & \text{if } t \in [0, t_\mathrm{s}) \\ \operatorname{abs}\left(\tilde{G}_{x(i,\cdot)}\right)\operatorname{rad}\left(\overline{\Delta}(t)\right) & \text{if } t \in [t_\mathrm{s}, T] \end{cases} \qquad i = 1\ldots n_g\,,
$$

$$
\delta_{H_i} := \operatorname{abs}\left(H_{x(i,\cdot)}\right)\ \operatorname{rad}\left(\overline{\Delta}(T)\right)\,, \quad i = 1\ldots n_h\,.
$$

4.4.2 Case of Ellipsoidal Tube Enclosures

Under the assumption that the time-varying uncertainty $w(t)$ is bounded within the ellipsoid $\mathcal{E}(\hat{w}(t), Q_w(t))$, an ellipsoidal tube enclosure $\bar{\Delta}(t) := \mathcal{E}(Q_x(t)) \supseteq \Delta(t)$, parameterized by the matrix-valued function $Q_x : [0,T] \to \mathbb{S}_+^{n_x}$, can be precomputed by solving the auxiliary ODEs:

$$
\begin{aligned}
& \text{a. e. } t \in [0, t_s], \\
& \quad \dot{Q}_x(t) = F_x Q_x(t) + Q_x(t)F_x^\mathsf{T} + \kappa(t)Q_x(t) + \frac{1}{\kappa(t)}F_w Q_w F_w^\mathsf{T} \\
& \text{a. e. } t \in [t_s, T], \\
& \quad \dot{Q}_x(t) = \tilde{F}_x Q_x(t) + Q_x(t)\tilde{F}_x^\mathsf{T} + \kappa(t)Q_x(t) + \frac{1}{\kappa(t)}F_w Q_w F_w^\mathsf{T} \\
& \quad \text{with} \quad Q_x(0) = 0\,,
\end{aligned}
$$

for any given function $\kappa : [0,T] \to \mathbb{R}_{++}$. For instance, κ may be chosen to minimize the trace of $\dot{Q}_x(t)$,

$$
\kappa(t) := \sqrt{\frac{\operatorname{Tr}\left(F_w Q_w F_w^\mathsf{T}\right)}{\operatorname{Tr}\left(Q_x(t)\right) + \epsilon}}, \tag{4.12}
$$

for some finite tolerance $\epsilon > 0$.

Therefore, an alternative choice for the back-offs $\delta_G(t)$, $0 \leq t \leq T$, and δ_H in the robust counterpart problem (4.9) is

$$\delta_{G_i}(t) := \begin{cases} \sqrt{G_{x(i,\cdot)} Q_x(t) G_{x(i,\cdot)}^{\mathsf{T}}} & \text{if } t \in [0, t_{\mathrm{s}}) \\ \sqrt{\tilde{G}_{x(i,\cdot)} Q_x(t) \tilde{G}_{x(i,\cdot)}^{\mathsf{T}}} & \text{if } t \in [t_{\mathrm{s}}, T] \end{cases} \qquad i = 1 \ldots n_g \,,$$
$$\delta_{H_i} := \sqrt{H_{x(i,\cdot)} Q_x(t) H_{x(i,\cdot)}^{\mathsf{T}}} \,, \quad i = 1 \ldots n_h \,.$$

In practice, the choice of ellipsoidal reachable tubes instead of interval tubes may be dictated by the fact that the former are more efficient at mitigating the wrapping effect, thereby reducing the overall conservativeness. The case of a time-varying uncertainty $w(t)$ which is bounded within an interval vector $[w^{\mathrm{L}}(t), w^{\mathrm{U}}(t)]$ can be treated likewise, e.g. by regarding this interval vector as the Minkowski sum of n_w one-dimensional ellipsoids [136].

4.4.3 Selection of the Linear Gain Matrix

For the uncertain system to be ultimately bounded in $\bar{\Delta}(T)$, the enclosure of the reachable set should be robustly controlled invariant. For the continues-time system of (4.6) with uniqueness of solution, the set $\bar{\Delta}$ is robustly invariant for $d_x(t)$ if and only if

$$\dot{\delta} \in \mathcal{T}_{\overline{\Delta}}(\delta) \,, \quad \forall \delta = \partial \overline{\Delta} \,, \quad \forall w \in \mathcal{W} \tag{4.13}$$

where $\mathcal{T}_{\overline{\Delta}}(\delta)$ is the tangent cone to $\overline{\Delta}$ in δ, and the sub-tangentiality condition is meaningful only on the boundary. Note, the $\tilde{F}$ in (4.6) includes the effect of the closed-loop control, thus the closed and convex set $\overline{\Delta}$ is controlled invariant if there exist $u = u(\delta)$ such that the above condition is satisfied. And for the two special cases of polyhedral and ellipsoidal sets, sufficient conditions for the invariance can be satisfied when the sub-tangentiality condition on the vertices or the entire boundary, respectively, such that $\dot{\delta}$ points inside the set on the boundary.

The trade-off in the choice of the gain matrix K in (4.2) is that a larger gain will reduce the conservativeness inherent to a robust counterpart formulation, effectively approaching the performance of the nominal system in the limit, but this will come at the price of a more aggressive controller and thus a higher sensitivity to measurement noise.

In the case of an uncertain dynamic system, choosing K such that $[F_x + F_u K]$ is Hurwitz can prevent the reachable tube (4.7) from growing exponentially as time advances; but a stable tube may still be impractically large depending on the amount of uncertainty. By analogy to the target set in robust tube MPC [128, 131], a better choice for K instead is one for which the reachable set $\bar{\Delta}(T)$ is minimal robustly positively invariant (mRPI) for the linear system (4.6). In the case of ellipsoidal tubes, we may choose K as a (global) optimum of the problem

$$\begin{aligned}
\min_{K, Q_x} \quad & ||K||_2 \\
\text{s.t.} \quad & \text{a.e. } t \in [0, t_\text{s}), \\
& \quad \dot{Q}_x(t) = F_x Q_x(t) + Q_x(t) F_x^\mathsf{T} + \kappa(t) Q_x(t) + \frac{1}{\kappa(t)} F_w Q_w(t) F_w^\mathsf{T} \\
& \text{a.e. } t \in [t_\text{s}, T], \\
& \quad \dot{Q}_x(t) = \tilde{F}_x Q_x(t) + Q_x(t) \tilde{F}_x^\mathsf{T} + \kappa(t) Q_x(t) + \frac{1}{\kappa(t)} F_w Q_w(t) F_w^\mathsf{T} \\
& Q_x(0) = 0, \quad \dot{Q}_x(T) \preceq 0\,,
\end{aligned} \tag{4.14}$$

with the weights $\kappa(t)$ as in (4.12) above. Notice that such problems are nonconvex in general, thus rendering the determination of K a difficult task. A further complication in the case of interval tubes is that the dynamic may be nonsmooth.

4.4.4 Extension to Multiplicative Uncertainty

As an extension to the additive uncertainty case in problem (4.1), we consider the broader class of uncertain dynamic systems with uncertain state coefficient matrix such that

$$\dot{x}(t) = (F_x + \mathcal{F}(t))x(t) + F_u u(t) + F_\theta \theta + F_w w(t) + f_0\,,$$

where the matrix $\mathcal{F}(t) := \sum_{j=1}^{p} P^j \omega_j(t)$ are uncertain, with some given scaling matrices $P^1, \ldots, P^p$. Notice that a further extension to problems where the matrices F_u and F_θ are also affected by uncertainty is also possible, for instance by invoking similar arguments as in [125].

Applying a similar splitting of the state and the disturbance into nominal and perturbed components as in (4.3) and define

$$\mathcal{F}(t) = \hat{\mathcal{F}} + d_{\mathcal{F}}(t)$$

with $\hat{\mathcal{F}} = \sum_{j=1}^{p} P^j \hat{\omega}_j$ as its nominal part and $d_{\mathcal{F}}(t) = \sum_{j=1}^{p} P^j d_{\omega_j}(t)$ as its perturbed part. For simplicity of exposition and without loss of generality, we shall assume here that the uncertainty is bounded within the unit ball, $d_\omega(t)^\mathsf{T} d_\omega(t) \leq 1$, i.e. $d_\omega \in \mathcal{E}(I_p)$.

The dynamics of state reference $\hat{x}$ become

$$\begin{aligned}
&\dot{\hat{x}}(t) = (F_x + \hat{\mathcal{F}})\hat{x}(t) + F_u \hat{u}(t) + F_\theta \theta + F_w \hat{w} + f_0 \\
&\text{with} \quad \hat{x}(0) = B_0 \theta + b_0 \, .
\end{aligned}$$

The dynamics of the state perturbation d_x become

$$\begin{aligned}
&\dot{d}_x(t) = \begin{cases} (F_x + \hat{\mathcal{F}}) d_x(t) + d_{\mathcal{F}}(t)(\hat{x}(t) + d_x(t)) + F_w d_w(t) & \text{if } t \in [0, t_\mathrm{s}) \\ (\tilde{F}_x + \hat{\mathcal{F}}) d_x(t) + d_{\mathcal{F}}(t)(\hat{x}(t) + d_x(t)) + F_w d_w(t) & \text{if } t \in [t_s, T] \end{cases} \qquad (4.15) \\
&\text{with} \quad d_x(0) = 0 \, ,
\end{aligned}$$

and are now dependent on the control u and the parameter θ via the nominal state trajectory $\hat{x}$. Therefore, a similar strategy as for additive uncertainty, whereby reachability tubes can be precomputed for the state disturbances, will rely on the availability of a conservative enclosure $\hat{X}(t) \in \mathbb{K}^{n_x}$ for the nominal state. For instance, such enclosures can be obtained using state-of-the-art set-valued integrators [80, 136]. The matrix $\tilde{F}$ takes the same form as in the case of additive uncertainty, and the feedback gain K can be computed using the same equations.

Here, an interval tube enclosure $\overline{\Delta}(t) := \left[\delta_x^\mathrm{L}(t), \delta_x^\mathrm{U}(t)\right] \supseteq \Delta(t)$ can be precomputed by solving the auxiliary ODEs:

$$
\begin{aligned}
&\text{a. e. } t \in [0,t_\mathrm{s}), \quad \forall i \in 1,\dots,n_x,\\
&\dot{d}^{\mathrm{L}}_{x_i}(t) = \min_{\delta_x,\delta_w,\xi} \left\{ \begin{array}{l|l}
\left[F_{x(i,\cdot)} + \sum_{j=1}^{p} P^j_{(i,\cdot)}\hat{\omega}_j\right]\delta_x & \delta_{x_i} = d^{\mathrm{L}}_{x_i}(t),\\
+ \sum_{j=1}^{p} P^j_{(i,\cdot)}\delta_{\omega_j}[\hat{\xi} + \delta_x] & \delta_x \in \bar{\Delta}(t),\\
+ F_w\delta_w & \delta_\omega \in \mathcal{E}(I_p),\\
 & \delta_w \in W \ominus \{\hat{w}(t)\},\\
 & \xi \in \hat{X}
\end{array}\right\}\\
&\dot{d}^{\mathrm{U}}_{x_i}(t) = \max_{\delta_x,\delta_w,\xi} \left\{ \begin{array}{l|l}
\left[F_{x(i,\cdot)} + \sum_{j=1}^{p} P^j_{(i,\cdot)}\hat{\omega}_j\right]\delta_x & \delta_{x_i} = d^{\mathrm{U}}_{x_i}(t),\\
+ \sum_{j=1}^{p} P^j_{(i,\cdot)}\delta_{\omega_j}[\hat{\xi} + \delta_x] & \delta_x \in \bar{\Delta}(t),\\
+ F_w\delta_w & \delta_\omega \in \mathcal{E}(I_p),\\
 & \delta_w \in W \ominus \{\hat{w}(t)\},\\
 & \xi \in \hat{X}
\end{array}\right\}\\
&\text{a. e. } t \in [t_\mathrm{s},T], \quad \forall i \in 1,\dots,n_x,\\
&\dot{d}^{\mathrm{L}}_{x_i}(t) = \min_{\delta_x,\delta_w,\xi} \left\{ \begin{array}{l|l}
\left[\tilde{F}_{x(i,\cdot)} + \sum_{j=1}^{p} P^j_{(i,\cdot)}\hat{\omega}_j\right]\delta_x & \delta_{x_i} = d^{\mathrm{L}}_{x_i}(t),\\
+ \sum_{j=1}^{p} P^j_{(i,\cdot)}\delta_{\omega_j}[\hat{\xi} + \delta_x] & \delta_x \in \bar{\Delta}(t),\\
+ F_w\delta_w & \delta_\omega \in \mathcal{E}(I_p),\\
 & \delta_w \in W \ominus \{\hat{w}(t)\},\\
 & \xi \in \hat{X}
\end{array}\right\}\\
&\dot{d}^{\mathrm{U}}_{x_i}(t) = \max_{\delta_x,\delta_w,\xi} \left\{ \begin{array}{l|l}
\left[\tilde{F}_{x(i,\cdot)} + \sum_{j=1}^{p} P^j_{(i,\cdot)}\hat{\omega}_j\right]\delta_x & \delta_{x_i} = d^{\mathrm{U}}_{x_i}(t),\\
+ \sum_{j=1}^{p} P^j_{(i,\cdot)}\delta_{w_j}[\hat{\xi} + \delta_x] & \delta_x \in \bar{\Delta}(t),\\
+ F_w\delta_w & \delta_\omega \in \mathcal{E}(I_p),\\
 & \delta_w \in W \ominus \{\hat{w}(t)\},\\
 & \xi \in \hat{X}
\end{array}\right\}\\
&\quad \text{with} \quad d^{\mathrm{L}}_{x_i}(0) = d^{\mathrm{U}}_{x_i}(0) = 0\,.
\end{aligned} \tag{4.16}
$$

A matrix-valued function $Q_x : [0,T] \to \mathbb{S}^{n_x}_{+}$ describing an ellipsoidal tube enclosure of $\Delta(t)$ may be precomputed likewise. One approach involves determining ellipsoidal enclosures for the uncertainty-dependent terms in (4.15) using ellipsoidal calculus [139, 140],

$$\begin{aligned}&\forall (d_x, d_w, \hat{x}) \in \mathcal{E}(Q_x(t)) \times \Gamma \times \hat{X}(t),\\ d_{\mathcal{F}}(t)(\hat{x}(t) + d_x(t)) &+ F_w d_w(t) \in \mathcal{E}(\overline{Q}(t))\,,\end{aligned}$$

and then propagating Q_x through the auxiliary ODEs:

$$\begin{aligned}&\text{a. e. } t \in [0, t_s],\\ &\quad \dot{Q}_x(t) = (F_x + \hat{\mathcal{F}})Q_x(t) + Q_x(t)(F_x + \hat{\mathcal{F}})^\mathsf{T} + \kappa(t)Q_x(t) + \frac{1}{\kappa(t)}\overline{Q}(t)\\ &\text{a. e. } t \in [t_s, T],\\ &\quad \dot{Q}_x(t) = (\tilde{F}_x + \hat{\mathcal{F}})Q_x(t) + Q_x(t)(\tilde{F}_x + \hat{\mathcal{F}})^\mathsf{T} + \kappa(t)Q_x(t) + \frac{1}{\kappa(t)}\overline{Q}(t) \\ &\quad \text{with} \quad Q_x(0) = 0 \quad \text{and} \quad \kappa(t) := \sqrt{\frac{\mathrm{Tr}(\overline{Q}(t))}{\mathrm{Tr}\,(Q_x(t)) + \epsilon}}\,.\end{aligned} \tag{4.17}$$

Another, potentially tighter, approach to propagating Q_x is based on the recent work on linear control system with multiplicative uncertainty by [141]. Assume a general ellipsoidal enclosure for (4.15) is in the form:

$$\begin{aligned}&\dot{Q}_x(t) = \begin{cases} (F_x + \hat{\mathcal{F}})Q_x(t) + Q_x(t)(F_x + \hat{\mathcal{F}})^\mathsf{T} + \Omega_t(t, Q_x, \Gamma, \hat{X}) & \text{if } t \in [0, t_\mathrm{s})\\ (\tilde{F}_x + \hat{\mathcal{F}})Q_x(t) + Q_x(t)(\tilde{F}_x + \hat{\mathcal{F}})^\mathsf{T} + \Omega_t(t, Q_x, \Gamma, \hat{X}) & \text{if } t \in [t_s, T]\end{cases}\\ &\text{with} \quad Q_x(0) = 0\end{aligned} \tag{4.18}$$

For the multiplication of two terms with ellipsoidal uncertainty, such that

$$x^+ = \mathcal{P}x\,,$$

where $\mathcal{P} \in \left\{\sum_{j=1}^{p} P_i\delta_i \middle| \delta \in \mathbb{R}^p, \delta^\mathsf{T}\delta \leq 1\right\}$ with some scaling matrices P_i and $x \in \mathcal{E}(Q)$, an ellipsoidal enclosure for $x^+ \in \mathcal{E}(Q^+)$ can be computed using [141]

$$Q^+ = P(I \otimes S^{-1})P^\mathsf{T} \tag{4.19}$$

for any positive definite scaling matrix $S \in \mathbb{S}_{++}$ with $I - Q^{\frac{1}{2}}SQ^{\frac{1}{2}} \succ 0$, and $P = [P_1, \ldots, P_p]$.

Moreover, for the case of continuous-time linear dynamics with multiplication of ellipsoidal uncertainty

$$\dot{\xi}(t) = (F + \mathcal{P})\xi(t) \quad \text{with} \quad x(0) = 0$$

where the uncertainty set $\mathcal{P} \in \left\{ \sum_{j=1}^{p} P_i \delta_i \middle| \delta \in \mathbb{R}^p, \delta^\mathsf{T}\delta \leq 1 \right\}$ with some scaling matrices P_i and $\xi \in \mathcal{E}(Q)$, an ellipsoidal enclosure for the state of the uncertain dynamic system, such that $\xi(t) \in \mathcal{E}(Q(t))$, can be computed as

$$\dot{Q}(t) = FQ(t) + Q(t)F^\mathsf{T} + Q(t)S^{-1}Q(t) + P\left(I \otimes S\right)P^\mathsf{T}$$

for any $S \succ 0$ and $P = [P_1, \ldots, P_p]$.

Rewrite the dynamics for disturbed component (4.6) as in the form

$$\dot{d}_x(t) = \begin{cases} (F_x + \hat{\mathcal{F}} + d_{\mathcal{F}})d_x(t) + d_{\mathcal{F}}(t)\hat{x}(t) + F_w d_w(t) & \text{if } t \in [0, t_\mathrm{s}) \\ (\tilde{F}_x + \hat{\mathcal{F}} + d_{\mathcal{F}})d_x(t) + d_{\mathcal{F}}(t)\hat{x}(t) + F_w d_w(t) & \text{if } t \in [t_s, T] \end{cases}$$

$$\text{with} \quad d_x(0) = 0\,,$$

An ellipsoidal enclosure for the disturbed component can be computed by taking Ω_t in (4.18) as

$$\begin{aligned} \Omega_t(Q_x, \Gamma, \hat{X}) = {} & Q_x(t)S^{-1}Q_x(t) + P\left(I \otimes S\right)P^\mathsf{T} \\ & + \kappa Q_x(t) + \frac{1}{\kappa}Q_a + \eta Q_x(t) + \frac{1}{\eta}F_w Q_w F_w^\mathsf{T} \end{aligned} \tag{4.20}$$

for $P = [P^1, \ldots, P^p]$, any $S \succ 0$, $\kappa > 0$ and $\eta > 0$. Here, the parameters $S \succ 0$, $\kappa > 0$ and $\eta > 0$ are chosen such that

$$\begin{aligned} & S := \arg\min \mathrm{Tr}\left(Q_x(t)S^{-1}Q_x(t) + P\left(I \otimes S\right)P^\mathsf{T}\right)\,, \\ & \kappa(t) := \sqrt{\frac{\mathrm{Tr}\left(Q_a\right)}{\mathrm{Tr}\left(Q_x(t)\right) + \epsilon}}\,, \quad \text{and} \quad \eta(t) := \sqrt{\frac{\mathrm{Tr}\left(F_w Q_w F_w^\mathsf{T}\right)}{\mathrm{Tr}\left(Q_x(t)\right) + \epsilon}}\,. \end{aligned}$$

Here, Q_a in (4.20) represents the enclosure for $d_{\mathcal{F}}(t)\hat{x}(t)$ computed following (4.19) such that

$$Q_a = P(I \otimes \tilde{S}^{-1})P^\mathsf{T}$$

for any positive definite scaling matrix $\tilde{S} \in \mathbb{S}_{++}$ with $I - Q_{\hat{x}}^{\frac{1}{2}} \tilde{S} Q_{\hat{x}}^{\frac{1}{2}} \succ 0$, where $\tilde{S}$ is chosen to minimize $\mathrm{Tr}\,(Q_a)$.

Finally, the gain matrix K can be computed similar as in (4.14) except that the dynamics of $Q_x(t)$ propagates by (4.18).

4.5 Numerical case studies

In this section, we demonstrate the implementation of robust mp-NCO-tracking controller for an FCC unit and a system with two CSTRs in series.

4.5.1 An FCC system

Here, we continue with a fluidized catalytic cracking (FCC) unit system as in Section. 2.4.3. And the uncertainty is considered in two cases, i.e. additive and multiplicative disturbances. The objective is to steer the system to a given operating point, defined in terms of the mass fraction of coke on regenerated catalyst, C_{rc}, and the regenerator dense bed temperature, T_{rg}. The manipulated variables are the flow rate of air sent to the regenerator, F_{a}, and the catalyst flow rate, F_{s}. A linear input-output dynamic model is obtained via linearization and reduction of a first-principles nonlinear model around the equilibrium point $C^*_{\mathrm{rc}} = 5.207 \times 10^{-3}$, $T^*_{\mathrm{rg}} = 965.4\,\mathrm{K}$, $T^*_{\mathrm{ro}} = 776.9\,\mathrm{K}$, $T^*_{\mathrm{cy}} = 988.1\,\mathrm{K}$, and $T^*_{\mathrm{f}} = 400\,\mathrm{K}$, where T_{f} denotes the feed oil temperature. The control and state variables in this linear dynamic system are

$$x(t) := \begin{bmatrix} C_{\mathrm{rc}}(t) - C^*_{\mathrm{rc}} \\ T_{\mathrm{rg}}(t) - T^*_{\mathrm{rg}} \end{bmatrix}, \qquad u(t) := \begin{bmatrix} F_{\mathrm{s}}(t) - F^*_{\mathrm{s}} \\ F_{\mathrm{a}}(t) - F^*_{\mathrm{a}} \end{bmatrix}.$$

First, we consider the case with additive uncertainty, the feed oil temperature is taken as disturbances that varies around the nominal value. The corresponding optimization-based control problem is given by

$$
\begin{aligned}
\inf_{x,u} \quad & \frac{1}{2}\int_0^T x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t) \, \mathrm{d}t + \frac{1}{2} x(T)^\mathsf{T} Q_\mathrm{f} x(T) \\
\text{s.t.} \quad & \dot{x}(t) = (F_x + \mathcal{F})x(t) + F_u u(t) + F_w w(t) + f_0 \\
& g(x(t), u(t)) \leq 0 \\
& x(0) = \theta.
\end{aligned}
$$

with all matrices the same as in (2.34) except that

$$
F_w = \begin{bmatrix} 6.87 \times 10^{-7} \\ 2.47 \times 10^{-2} \end{bmatrix} .
$$

The path constraints $g(x(t), u(t))$ are given by:

$$
\begin{bmatrix} -100 \\ -15 \end{bmatrix} \leq u(t) \leq \begin{bmatrix} 100 \\ 15 \end{bmatrix} , \quad \begin{bmatrix} -10^{-3} \\ -20 \end{bmatrix} \leq x(t) \leq \begin{bmatrix} 10^{-3} \\ 20 \end{bmatrix} .
$$

The parameters θ correspond to uncertainty in the initial values of C_rc and T_rg, with $\Theta = [-10^{-3}, 10^{-3}] \times [-20, 20]$. And we consider the feed oil temperature as the additive disturbance is $w(t) \in W := [-10, 10]$, with reference value $\hat{w} = 0$. Despite a stable state coefficient matrix $F_x \prec 0$, the cross-section of the open-loop reachability tube enclosures in Figs. 4.2 & 4.3 grow significantly over time due to the cumulated uncertainty. The gain matrix K^* is computed by the optimization problem (4.14) with $t_s = 1$ so that $\bar{\Delta}(t)$ is mRPI at $t = 10$. The corresponding reachability tube enclosure in Fig. 4.4(a) is significantly smaller. An intermediate tube enclosure with feedback gain taken as $0.2K^*$ is also shown in Fig. 4.4(b) for the sake of comparison.

Having precomputed the reachability tube enclosures for a given gain matrix K, the next step entails mapping the critical regions in the robust mp-DO counterparts. The plots in Fig. 4.5 compare the critical regions of the robust mp-DO problems to those of the nominal mp-DO problem ($F_w = 0$), for the gain matrices K^* and $0.2K^*$ and ellipsoidal tube enclosures. Eleven regions are obtained in both cases, as described in Section. 2.4.3. By setting up a fully connected neural network $\mathcal{C}_\theta$ with two hidden layers and each layer containing twenty neurons with $\tanh(x) = (1 - e^{-2x})/(1 + e^{-2x})$ as activation function in the hidden layer and Softmax classifier for output layer, and training it until a testing

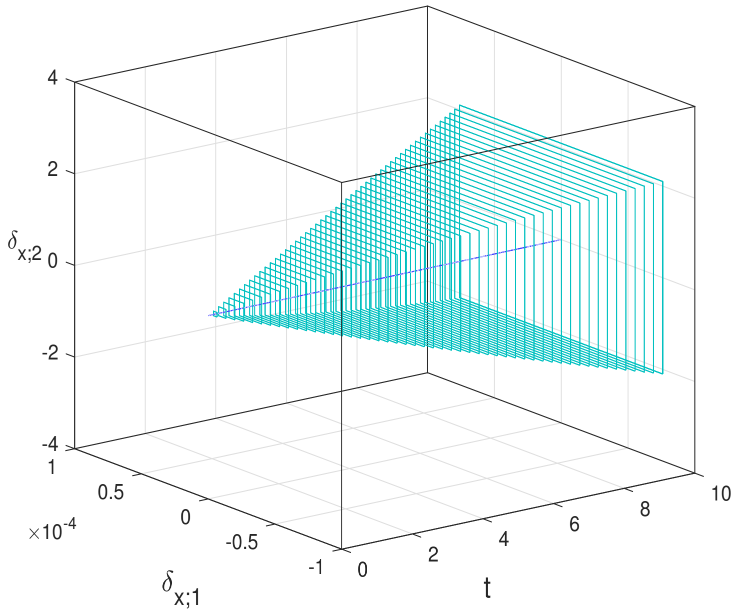


Figure 4.2: Interval reachability tubes without K – Case of additive disturbance.

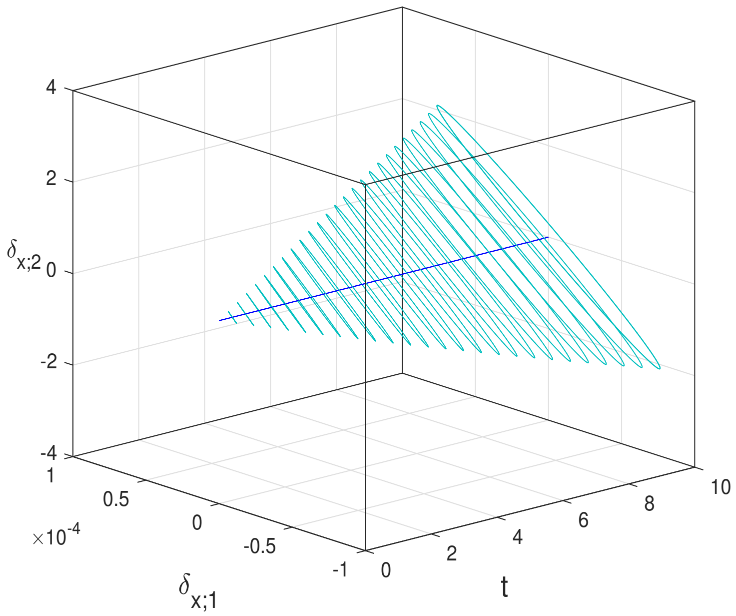
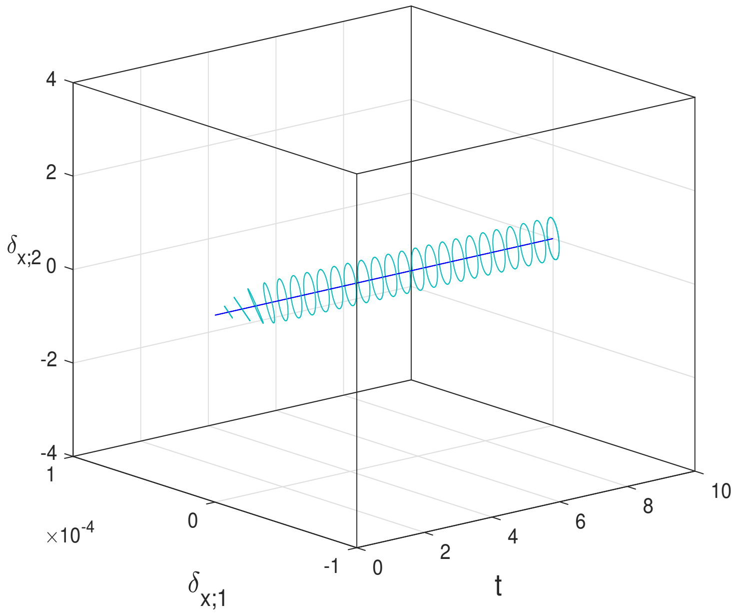
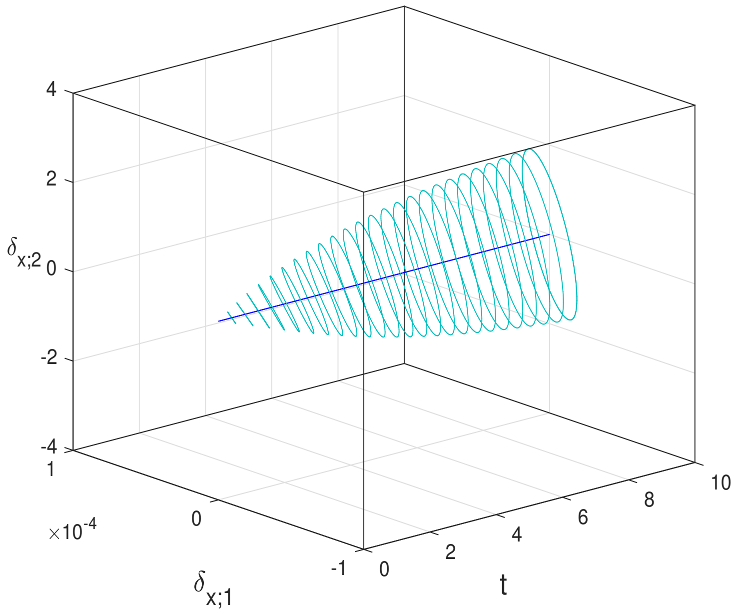


Figure 4.3: Ellipsoidal reachability tubes without K – Case of additive disturbance.

accuracy of $\hat{\epsilon} = 99.9\%$ based on the highest probability for evaluation and testing is achieved. By using a testing set of 1000 points with uniform distribution, the actually accuracy $\epsilon = 99.7\%$ is accepted with confidence of 95% based on the prediction of highest probability. The discrepancy between robust case and nominal case is reduced as the feedback gain K increases. With the reduced gain $0.2K^*$, the main effect of the back-offs is seen in the lower-left and upper-right corners, where the regions CR_4, CR_5, CR_9 and CR_{10}, and to a lesser extend the regions CR_3, CR_6, CR_8 and CR_{11}, are enlarged due to the state constraints being tightened. The solutions structures can be kept almost the same, but the input can be varied a lot from the nominal value due to the existence of Kd_x.



(a) With K_1



(b) With $0.2K_1$

Figure 4.4: Ellipsoidal reachability tubes – Case of additive disturbance.

The effect of using different back-offs compared with solution without taking back-offs (nominal) are shown in Fig. 4.6, where the state x_1 is seen to remain away from its upper bound of 10^{-3}.The other plots with color lines denote using the feedback gains: with K^*, $0.2K^*$ and without K, respectively. The larger the feedback gain, the closer to the robustified reference stay around the nominal trajectory. Among the robust solutions, for case (K=0) the back-offs are too large such that the switching structure is changed to contain two arcs, compared to the other cases all with three arcs. With a larger gain, the back-offs are smaller and results are less conservative. Note, the inputs shown here are the reference solution $\hat{u}(t)$, whereas the closed-loop input trajectories may vary from this based on different realizations of disturbances, which can be seen in the following closed-loop simulations.

Closed-loop simulations by applying both nominal and robust mp-NCO-tracking controllers in MPC scheme with different time periods ΔT based on ellipsoidal reachability tube are shown in Fig. 4.7 - Fig. 4.9.

The closed-loop responses with $\Delta T = 1$ for a realization of disturbance jumps as $w(t) = 5 + 5H(t-2) - 15H(t-4) - 5H(t-6) + 10H(t-8) + 5H(t-10) + 5H(t-12) - 15H(t-14) - 5H(t-16) + 10H(t-18)$ with $H(t)$ as a unit step function, are shown in Fig. 4.7, for the cases as follows: (1) using nominal controllers without constraint back-off; (2-4) using robust controller based on K^*, $0.2K^*$ and without K; (5) using the nominal controllers assuming $w(t)$ can be measured, respectively. All scenarios eventually switch from CR_6 to CR_1. It can be seen that the robust controller keeps the response feasible at all times, whereas constraint violations exist with the nominal controller. Simulations with a larger $\Delta T = 2$ is shown in Fig. 4.8 for the case with nominal and robust controller with corresponding K^*. Compared with the case of $\Delta T = 1$, it can be seen that a smaller sampling time leads to less conservative responses, which is shown in Fig. 4.9 as comparison for the responses with K^* based on both robust controllers.

Finally, we illustrate the applicability of the methodology in the case of multiplicative uncertainty. In addition to the additive disturbance as in the previous case, multiplicative uncertainty is added with four P^j, $j \in [1, 4]$, each with one unique non-zero element taken as the value of corresponding element of $0.1|F_x|$. Here, the feedback gain K^* is computed as in the case of additive disturbances based on (4.14) with $t_s = 1$. A breakdown of the parameter space into 11 critical regions is obtained for the robust mp-DO problem with K^*, similar as shown in Fig. 4.5.

The pre-computed ellipsoidal reachability tubes are shown in Fig. 4.10, all centred at a zero reference trajectory. Here, ellipsoidal reachability tubes are found to be much less conservative than their interval counterparts nonetheless, due to the inability of the latter to handle the wrapping effect during the enclosure propagation. With feedback input, the tubes becomes smaller and stops growing. A set of robust control and response trajectories for the initial state $x(0) = [8 \times 10^{-4}, 14] \in \mathsf{CR}_6$ are shown in Fig. 4.11, where the back-offs are derived from ellipsoidal reachability tubes. The effect of using different back-offs, using feedback gains K^*, $0.2K^*$ and without K, are compared with solution without taking back-offs (nominal), where the state x_1 is seen to remain away from its upper bound of 10^{-3} for robust solutions.

Corresponding closed-loop simulations obtained by applying both the nominal and robust mp-NCO-tracking controllers with a sampling period of $\Delta T = 1$ are shown in Fig. 4.12, for a given random disturbance including both additive and multiplicative disturbances, where where the solution structures switch from CR_6 to CR_1. The cases shown are based on using nominal controllers without constraint back-off and using robust controller with feedback gain K^*, respectively. Like previously, the robust controller keeps the response feasible at all times, whereas the nominal controller fails to guarantee feasibility.

4.5.2 A series of chemical reactors

Consider the system of two well mixed, non-isothermal continuous stirred tank reactors CSTRs operating in series [123], as shown in Fig. 3.5, where three parallel irreversible elementary exothermic reactions of the form $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} U$, and $A \xrightarrow{k_3} R$ take place. A is the reactant species, B is the desired product and U and R are undesired by-products. The feed to the first reactor consists of flow rate F_0 with pure A, molar concentration C_{A0} and temperature T_0, and the output of the first reactor is fed to the second reactor. Another fresh feed to the second reactor consists of flow rate F_3 with pure A, molar concentration C_{A03} and temperature T_{03}. The mathematical model of the process derived from material and energy balances can be found in Section 3.4.2, where T, C_A, Q and V denote the temperature of the reactor, the concentration of species A, the rate fo heat input/removal from the reactor, and the volume of reactor, respectively, with subscript j denoting CSTR_j.

The control objective is to steer system to an unstable steady state $(C_{A1}^*, T_1^*, C_{A2}^*, T_2^*) = (3.59\ \mathrm{kmol/m^3}, 388.57\ \mathrm{K}, 2.55\ \mathrm{kmol/m^3}, 429.24\ \mathrm{K})$, with $Q_1^* = 0$ and $Q_2^* = 0$, while simultaneously maintaining reactant concentration and avoiding high temperatures in the presence of input actuations. Manipulated inputs are chose as F_0, Q_1, F_3, Q_2. A linear input-output dynamic model is obtained via linearization and reduction of a first-principles nonlinear model around the equilibrium point $(C_{A1}^*, T_1^*, C_{A2}^*, T_2^*)$. The control and state variables in this linear dynamic system are

$$x(t) := \begin{bmatrix} C_{A1}(t) - C_{A1}^* \\ T_1(t) - T_1^* \\ C_{A2}(t) - C_{A2}^* \\ T_2(t) - T_2^* \end{bmatrix}, \qquad u(t) := \begin{bmatrix} F_0(t) - F_0^* \\ Q_1(t) \\ F_3(t) - F_3^* \\ Q_2(t) \end{bmatrix}.$$

and the disturbance $w(t) := [C_{A0} - C_{A0}^*,\ T_0 - T_0^*,\ C_{A03} - C_{A03}^*,\ T_{03} - T_{03}^*]^{\mathrm{T}}$. The optimization-based control problem complies with the formulation in (4.1), and the matrices all take the same value as in Section 3.4.2 except that $F_w = \mathrm{diag}(4.998, 4.998, 10, 10)$.

The parameters θ correspond to uncertainty in the initial values of state variables $x(0)$, with $\theta \in \Theta := [-0.1, -20, -0.1, -20] \times [0.1, 20, 0.1, 20]$ and the additive disturbance $w(t)$ is time-varying in $W := [-0.1, -5, -0.1, -5] \times [0.1, 5, 0.1, 5]$ with $\hat{w} = 0$. The gain matrix K^* is computed by the optimization problem (4.14) with $t_s = 1$.

There are 17 critical regions for both the nominal case (with $F_w = 0$) as in 3.4.2 and the robust case (with $w(t) \in W$) with feedback gain K_3. Among all the critical regions, the active constrains include saturations of input u_4 and the first state x_1 reaching its lower and upper bounds. Detailed description of the switching structures for each critical region can is presented in Section 3.4.2. A classifier for the critical regions of θ with accuracy 99.64% with confidence 95% is trained, estimated from testing accuracy of 99.76% for a random Sobol set of 5000 points. And the network is based on a neural network with an input layer containing 4 neurons, five hidden layers each containing 20 neurons and an output layer containing 17 neurons.

The input and state reference trajectories originating form the initial state $x(0) = (-0.095, 19, 0.09, -18) \in \mathsf{CR}_8$ for the nominal case is shown in Fig. 4.13, where the worst-case trajectories based on reachability tube without using feedback in input is compared with the reference solution. It can be seen that, since the steady state is unstable, enclosures for disturbed state component will grow rapidly lead to infeasibility. Robust solution of optimal reference trajectory based on using feedback gain K^* is shown in Fig. 4.14, compared with the worst-case scenarios based on ellipsoidal enclosures for each input and state variable. Constraints back-offs for x_1 reaching its lower bound and u_4 reaching its upper bound can be observed, which coming from the effect of disturbed component $d_x(t)$. By using state feedback input, the system response lies within local region around the reference trajectory and remains feasible.

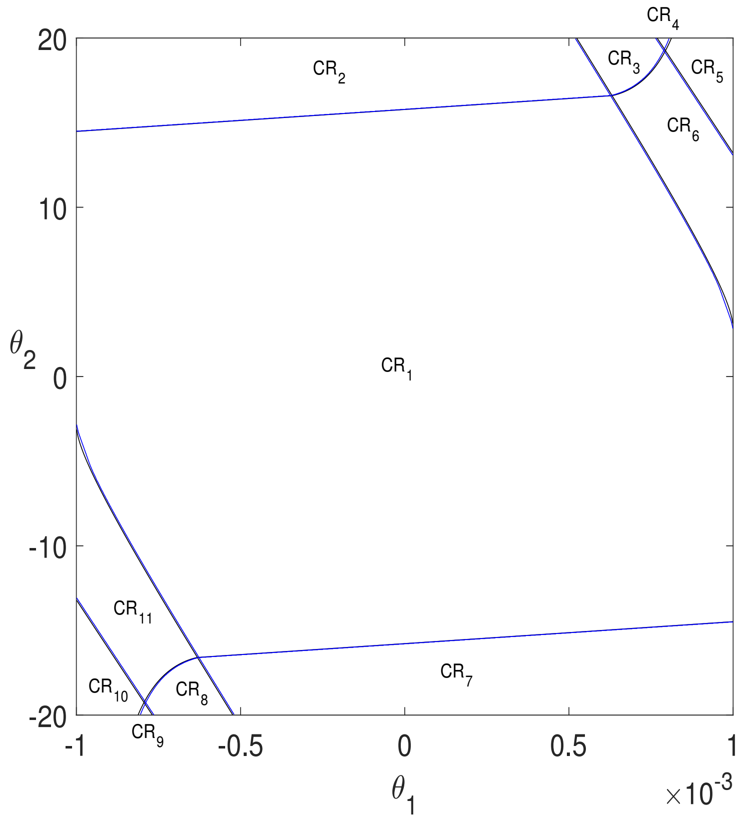
The closed-loop simulations for the initial state $x(0) = (-0.095, 19, 0.09, -18)$ by applying both the nominal and robust control with $\Delta T = 1$ for a realization of disturbances $w(t) = w_i \mathbf{1}_{[i,i+1)}(t)$ with $\mathbf{1}_A(t)$ as an indicator function, are shown in Fig. 4.15, where the sequence of w_i for disturbances in closed-loop simulation is $w_0 = [-0.05, -5, 0.05, 5]^{\mathrm{T}}$, $w_1 = [-0.05, 5, -0.05, 5]^{\mathrm{T}}$, $w_2 = [0.05, -5, -0.05, -5]^{\mathrm{T}}$, $w_3 = [0.05, 5, 0.05, -5]^{\mathrm{T}}$, $w_4 = [-0.05, -5, -0.05, -5]^{\mathrm{T}}$. The inputs are calculated based on the linear system and closed-loop responses are obtained by simulations of the original nonlinear dynamic system. The robust controller keeps the response feasible, but the nominal controller lead to serious constraint violations, x_1 exceeding its lower bound, which is plotted in grey dashed line in Fig. 4.15.

4.6 Conclusion

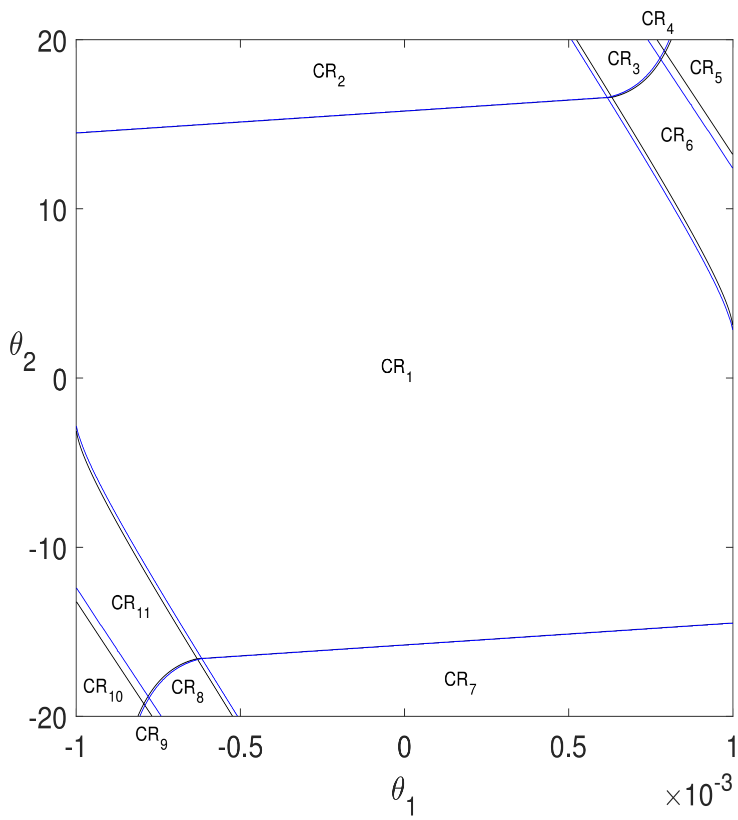
This chapter has presented an extension to the mp-NCO-tracking approach for the design of robust multi-parametric controllers for continuous-time linear dynamic systems subject to time-varying uncertainty. This extension involves backing-off the path and terminal state constraints based on a worst-case uncertainty propagation in the form of interval and ellipsoidal reachability tubes, where state feedback input is used for stability and reducing overestimation of the tubes.

An inherent advantage of the approach is that the back-offs can be computed prior to solving the mp-DO problem, thus enabling the direct application of the controller design procedure, as in Chapter 2, and making the off-line computational effort independent of the number of uncertain parameters. The off-line computation is based on the decomposition of the uncertain system state into two parts, namely, the nominal and disturbed components, where the dynamics of the disturbed component evolves separately from the reference trajectory and can be enclosed by interval or ellipsoidal reachability tubes. The robust counterpart of the control problem is then formulated, while inheriting the linearity from its nominal counterparts and retaining tractability of the nominal mp-NCO-tracking design problem.

The effect of backing-off the constraints is a modification of the size of the critical regions, and possibly the number of critical regions by either removing or adding extra regions. The applicability of the approach has been demonstrated by the case studies, considering either additive or multiplicative uncertainty. These developments are illustrated with the case studies of a fluidized catalytic cracking (FCC) unit operated in partial combustion mode and a system two CSTRs in series, where the applications of interval and ellipsoidal reachability tubes are compared, and the responses of different feedback gains are also studied in closed-loop simulations. By using the robust formulation for unknown disturbances, the robust controller keeps the system responses feasible at all times, whereas the nominal controller fails to guarantee feasibility.



(a) For K^*



(b) For $0.2K^*$

Figure 4.5: Critical regions of the mp-DO. Black lines: nominal regions; Blue lines: robust regions.

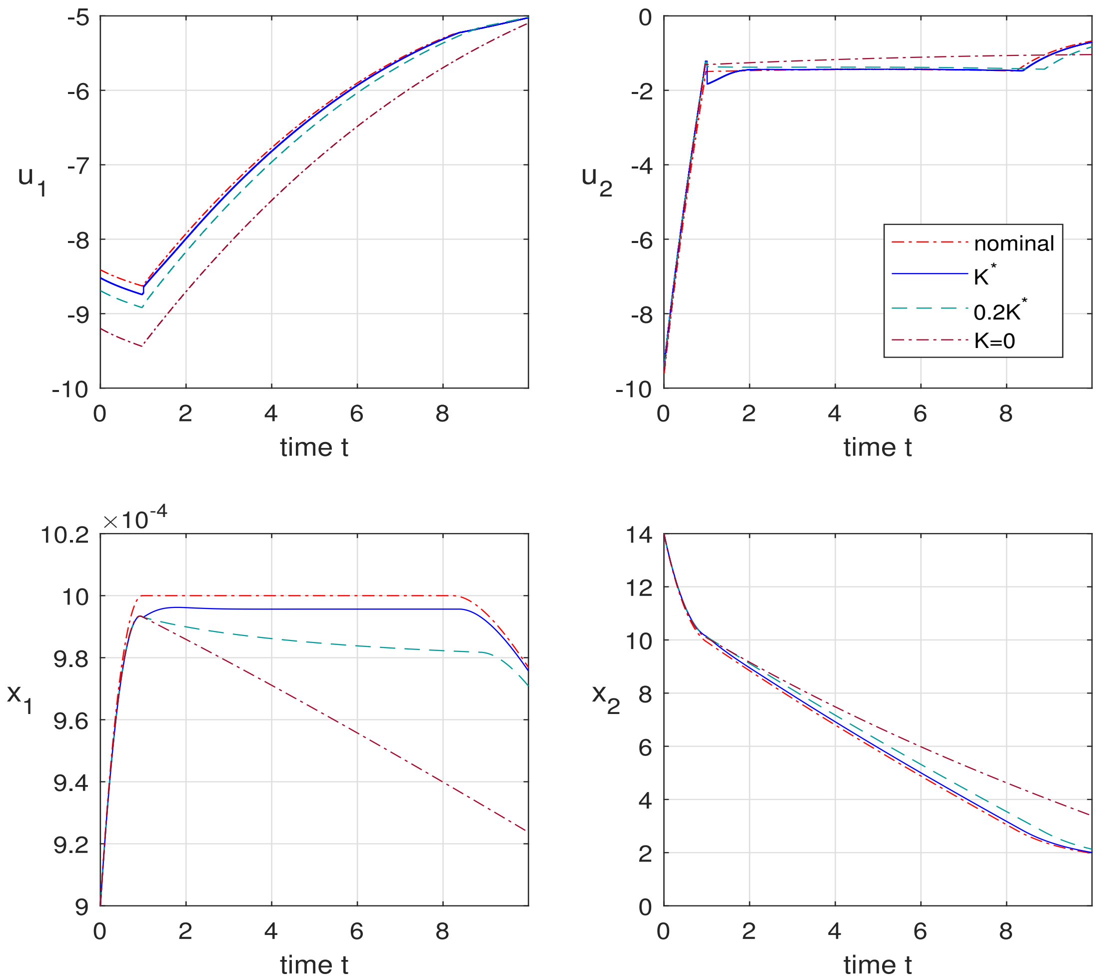


Figure 4.6: Robust mp-DO based on ellipsoidal reachability tubes – Case of additive disturbance.

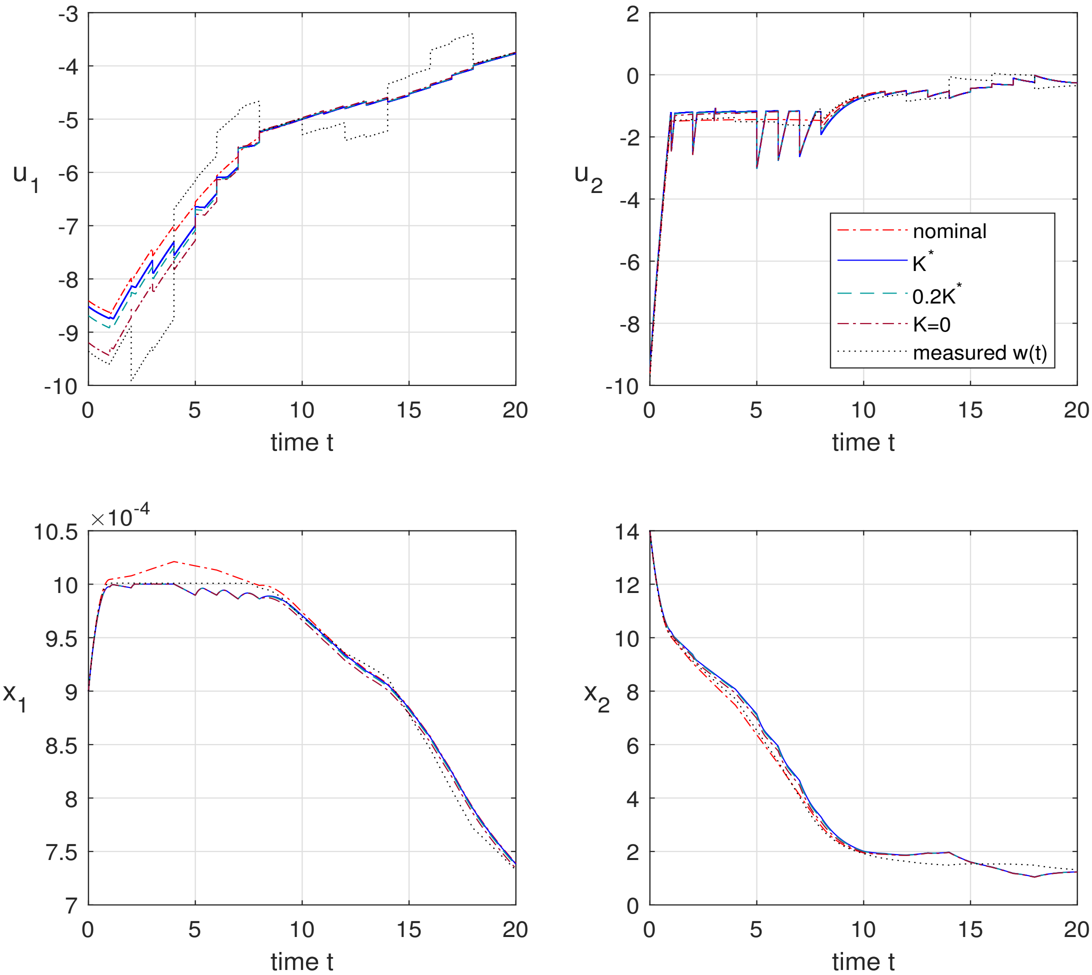


Figure 4.7: Closed-loop performance of the mp-NCO-tracking controllers with $\Delta T = 1$ – Case of additive disturbance.

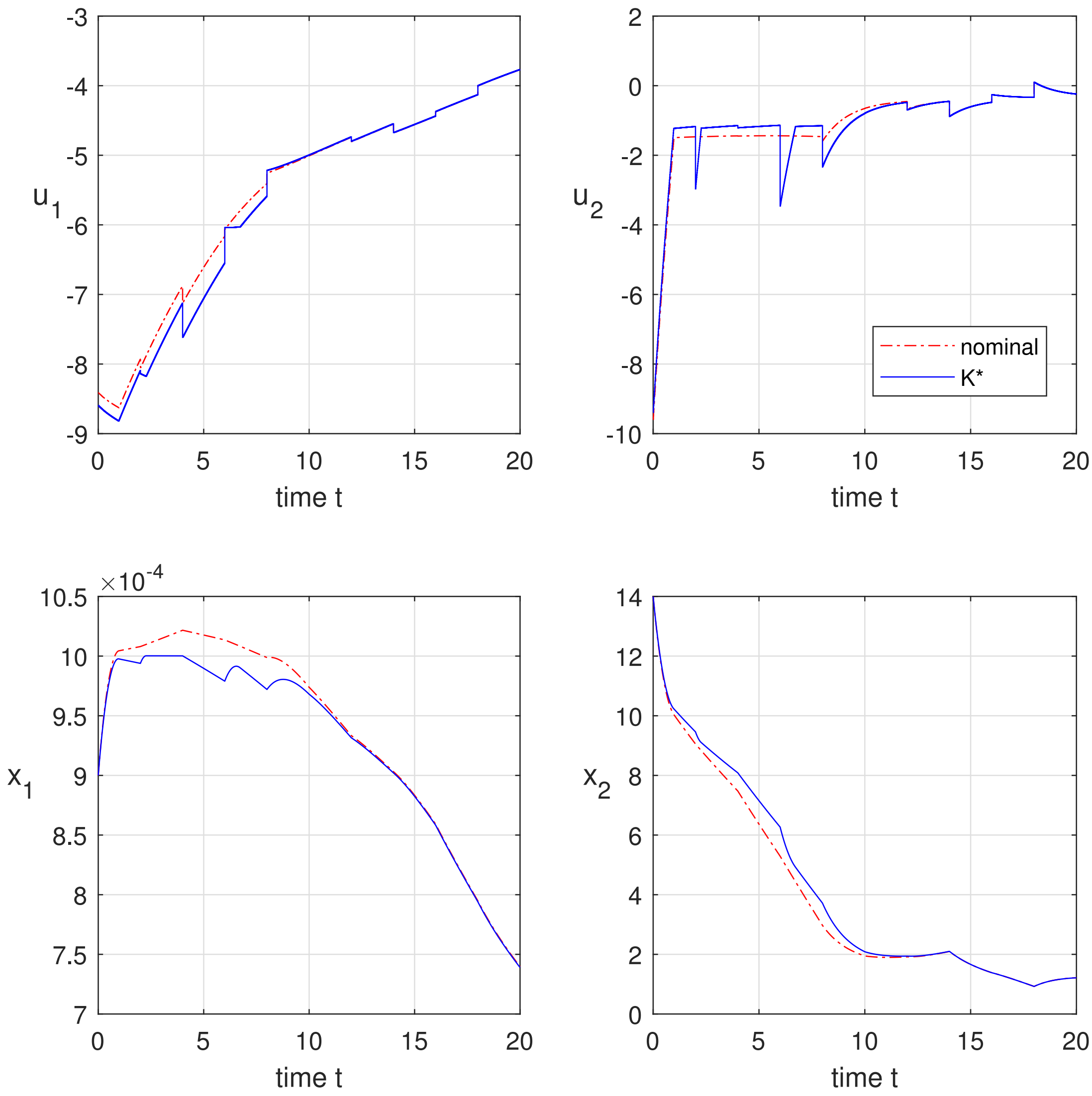


Figure 4.8: Closed-loop performance of the mp-NCO-tracking controllers with $\Delta T = 2$ – Case of additive disturbance.

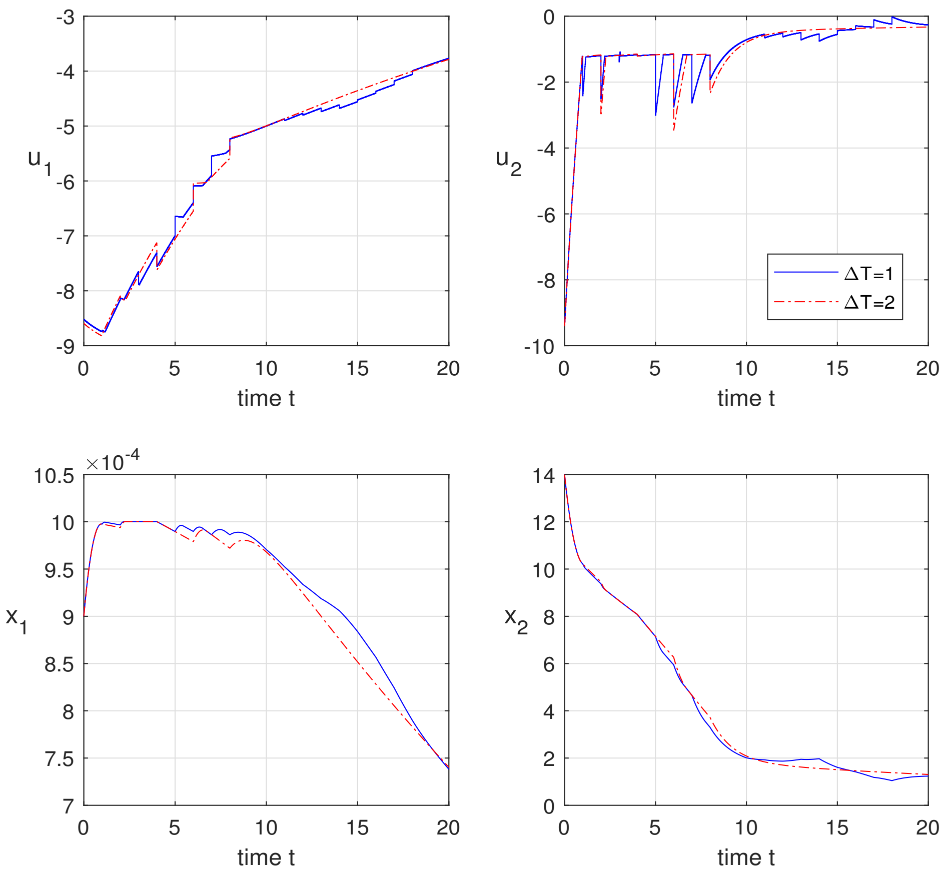
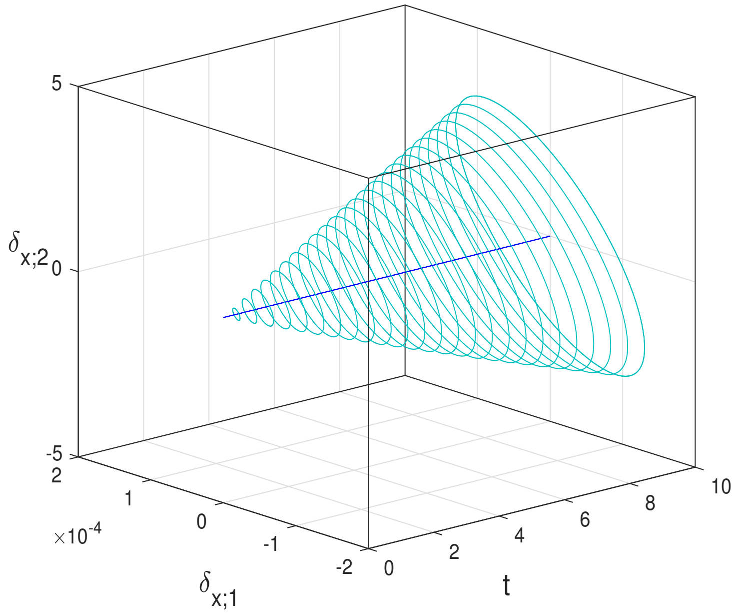
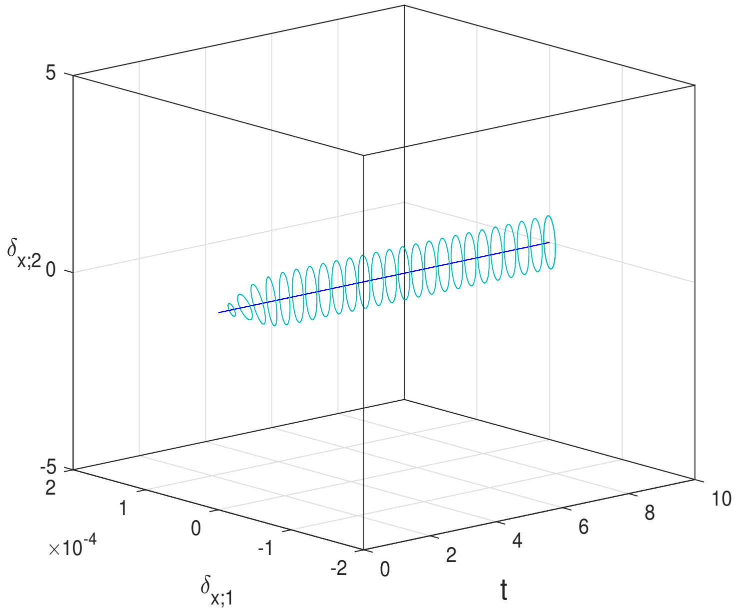


Figure 4.9: Comparison on closed-loop performance of the mp-NCO-tracking controllers – Case of additive disturbance.



(a) Ellipsoidal reachability tube without K



(b) Ellipsoidal reachability tube with K^*

Figure 4.10: Ellipsoidal reachability tubes – Case of multiplicative disturbance.

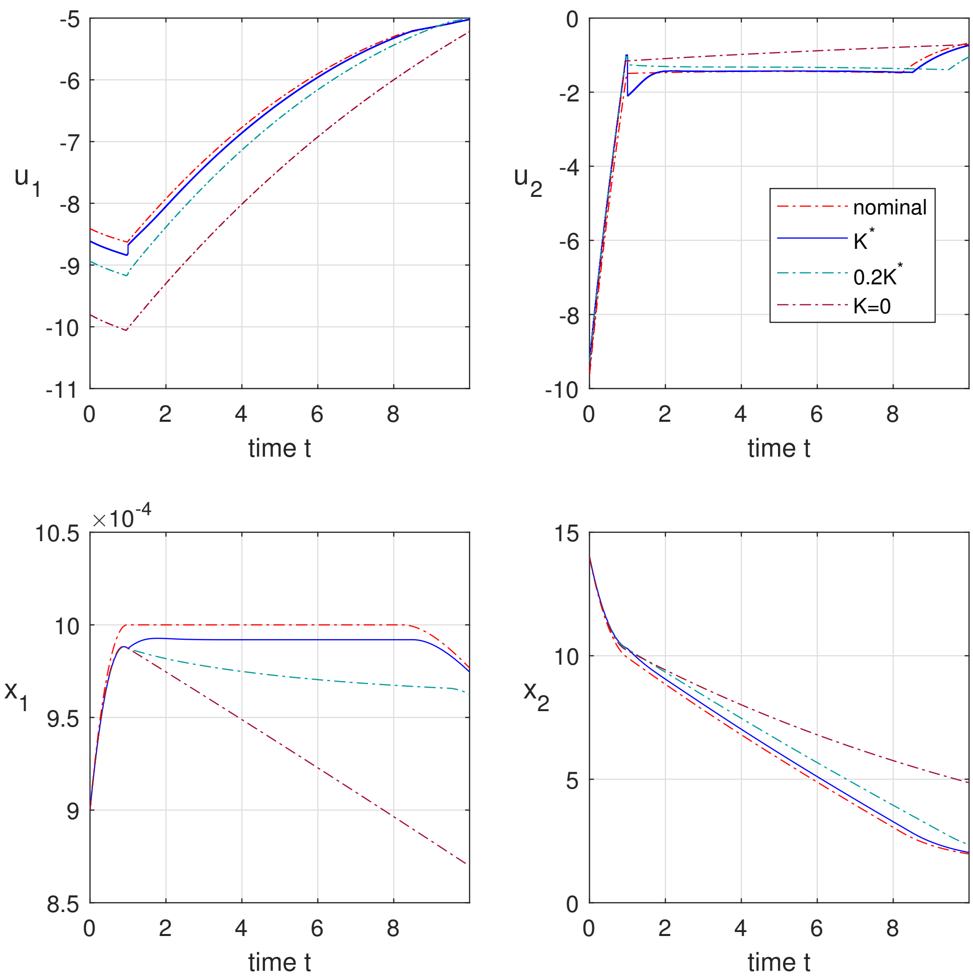


Figure 4.11: Robust mp-DO based on ellipsoidal reachability tubes – Case of multiplicative disturbance.

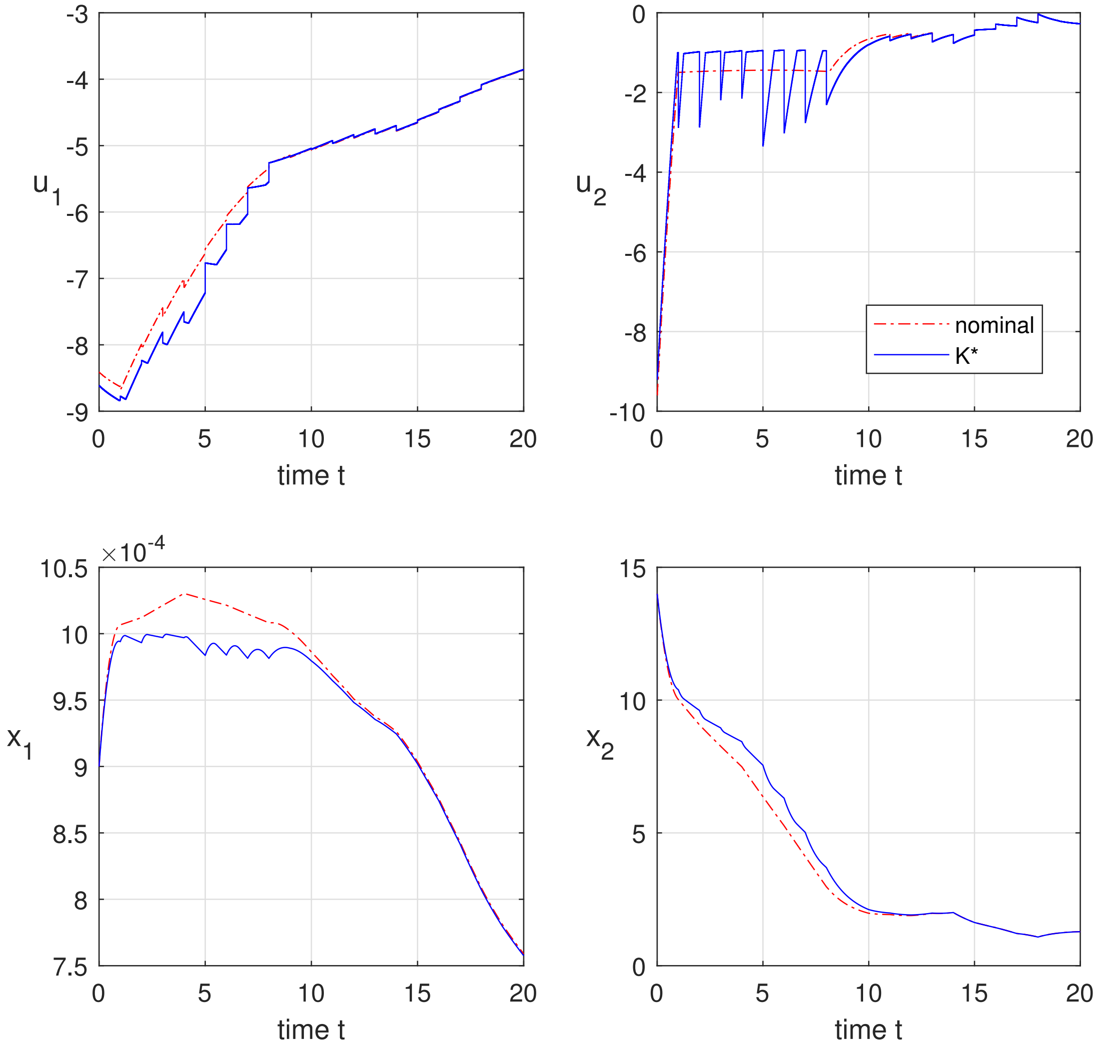


Figure 4.12: Closed-loop performance of the nominal and robust mp-NCO-tracking controllers – Case of multiplicative disturbance.

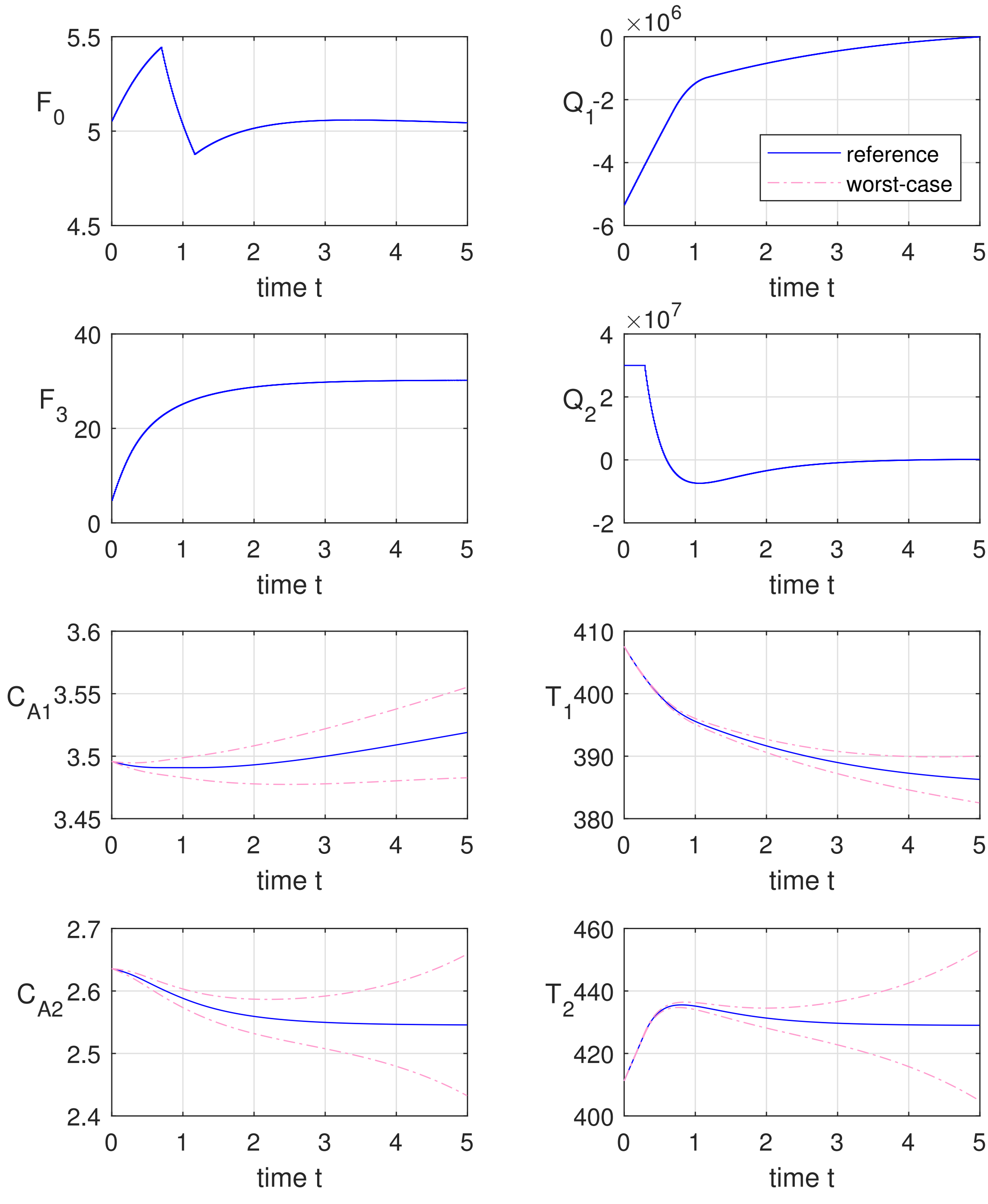


Figure 4.13: Nominal solution and worst case enclosure for CSTRs system with additive disturbance.

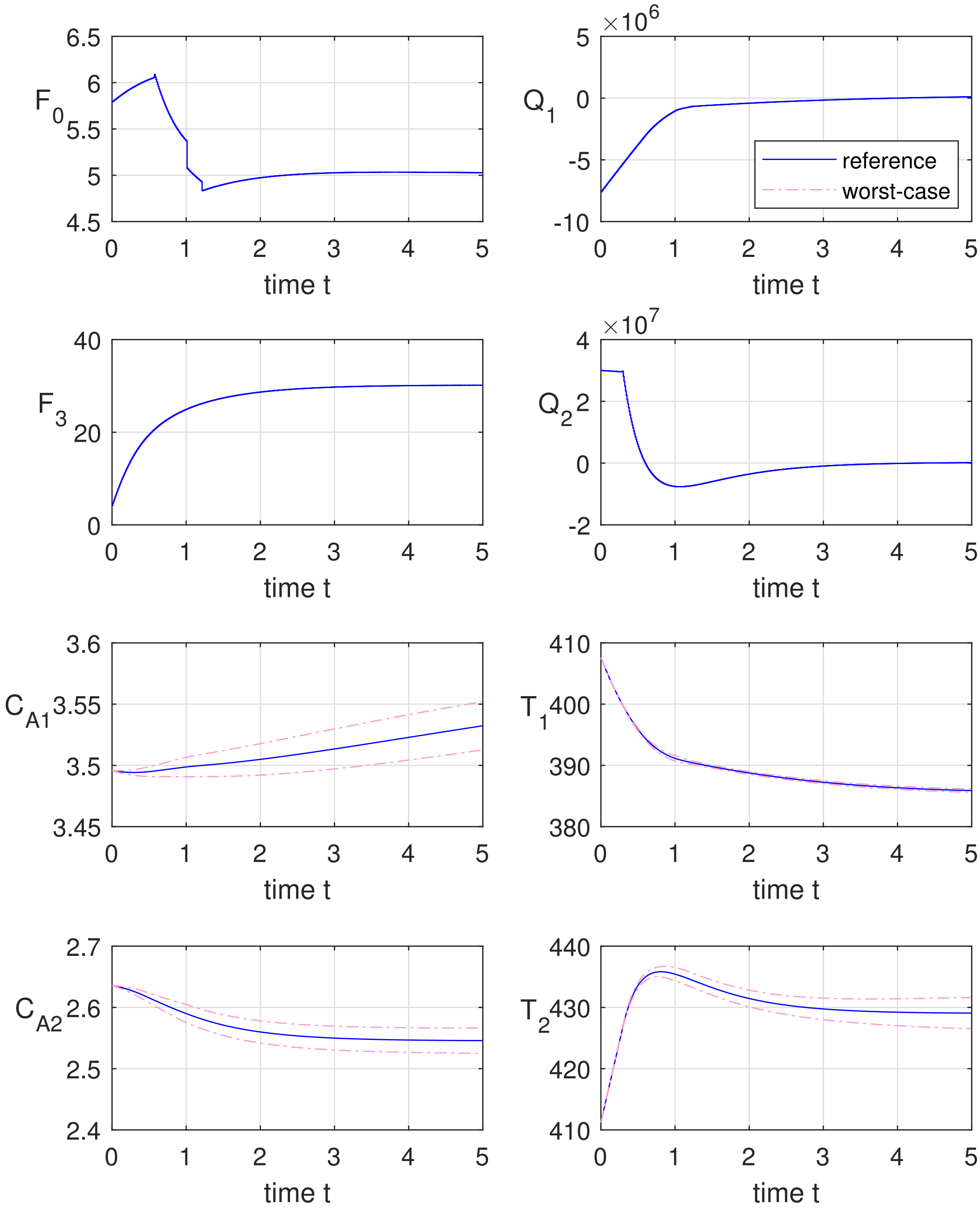


Figure 4.14: Robust mp-DO and worst case enclosure for CSTRs system with additive disturbance.

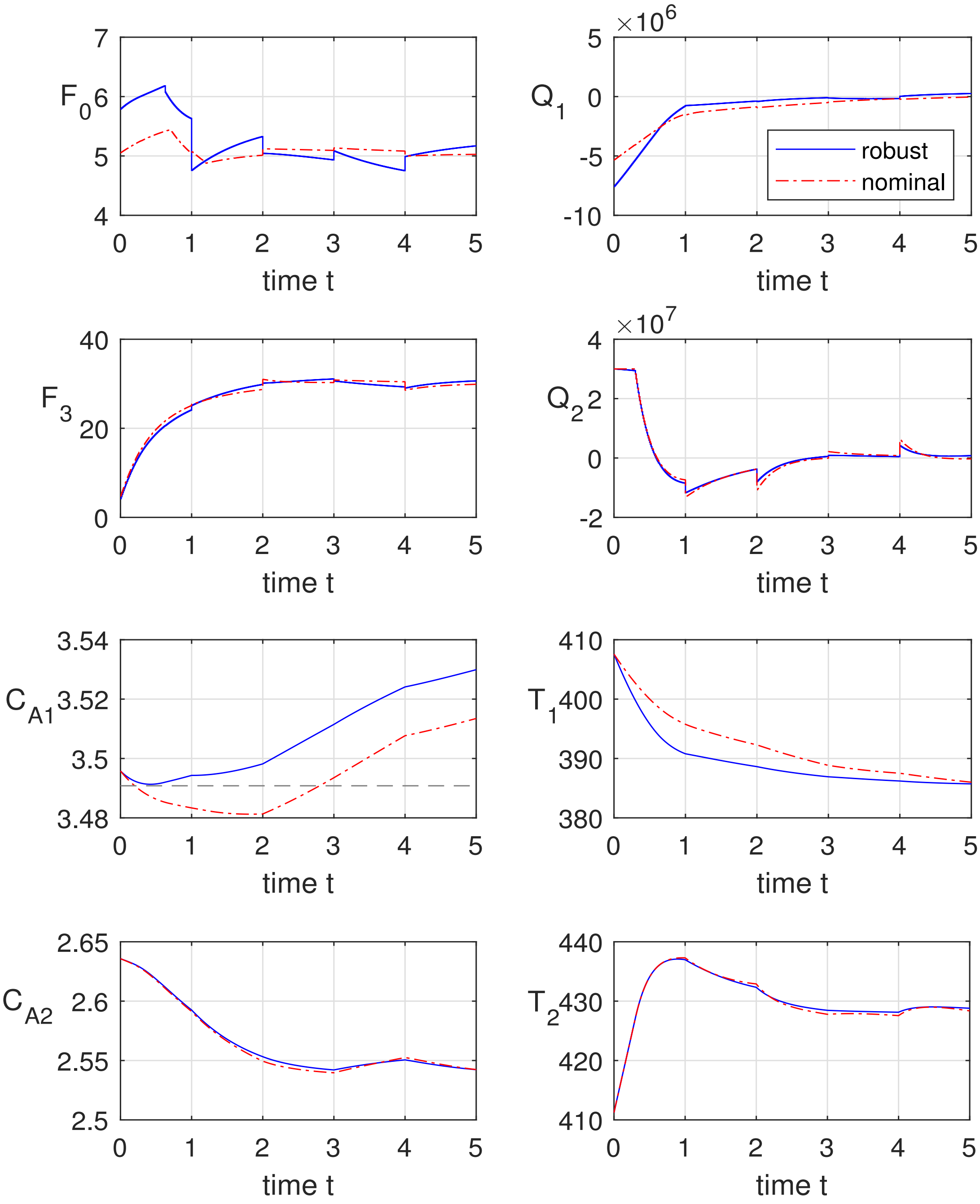


Figure 4.15: Closed-loop performance of the robust mp-NCO-tracking controllers for CSTRs system.

Chapter 5

Conclusions and future directions

In this thesis, the methodology of multi-parametric NCO-tracking has been proposed, by combining multi-parametric dynamic optimization and NCO-tracking for continuous-time linear dynamic systems. The multi-parametric solution structure of an optimal control problem with uncertainty for linear dynamic systems is explored, enabling relaxation of the fixed switching structure assumption in NCO-tracking and a dramatic reduction in the number of critical regions by the continuous-time form compared with classical mp-MPC controllers due to its ability to capture the underlying nonlinear feedback control nature.

An algorithm has been presented for characterizing the corresponding multi-parametric solution structure in terms of the exact critical regions and nonlinear feedback control laws, and the algorithm is readily applied in receding horizon control. The uncertain parameter domain is partitioned into a number of critical regions, each corresponding to a unique sequence of arcs, i.e., the sequence of active constraints or sensitivity conditions. The online application of the parametric controller involves determining the critical regions containing the measured parameter and applying the corresponding feedback law by the updated solution structure. By this approach, the optimal solution consists of different control switching structures and parameter jumps between different critical regions in the closed-loop control, which corresponds to changes of solution structures and has been illustrated in the numerical case studies.

For characterization of the critical regions, the boundary of which is generally nonlinear and nonconvex, we resort to the data-driven method for classification. By taking advantage of this well-studied area in machine learning, a classifier is trained offline for mapping a parameter into its corresponding critical region. The model of the classifier is constructed based on fully connected multi-layer neural networks for its representational power on non-linearities of arbitrary complexity. An algorithm is presented for adaptively training the classifier, which involves building a neural network model, generating the sample data and optimization-based training. The architecture of the classifier consists of an input layer of the same dimension as the uncertain parameter, multiple hidden layers used to mimic the continuous function denoting region boundaries, and an output layer with each neuron corresponding to one critical region. By using a softmax classifier in the output layer of the network to minimize the cross entropy between the prediction and ground truth labels, a normalized probability distribution for each critical region is obtained. The distribution is represented by a vector with each dimension corresponding to one critical region, and regions with highest scores are chosen as the candidates, which also manages the cases of misclassification. There exist numerous choices for training the classification in the existing publications by which the model can be built and trained. Once the training process is completed, the training set can be completely discarded, and locating the parameter only requires very simple function evaluations. The entire procedure is illustrated by the case studies of an FCC system and a series of chemical reactors, in which classifiers with an estimated accuracy with certain confidence based on binomial distribution are applied.

Finally, a robust mp-NCO-tracking approach is presented by extending the multi-parametric controller for continuous-time linear dynamics systems subject to bounded additive and multiplicative time-varying uncertainty. The uncertain system state is decomposed into two components, namely, the nominal and disturbed components, which enables splitting the system dynamics into two parts. The disturbance only exists in the dynamics of the disturbed component, which evolves via some auxiliary ODEs separately from the nominal part. Thus, a convex enclosure can be computed using interval or ellipsoidal reachability tubes by implementation of the continuous-time set propagation methods. As a result, the uncertain system state is enclosed by a continuous-time reachability tube centred at the nominal reference trajectory. The robust counterpart of the control problem is then formulated by backing off the path and terminal constraints based on a worst-case uncertainty propagation, in which the support functions for interval or ellipsoidal reachability tubes are used for computing the back-offs for the constraints. Furthermore, additional state feedback input can be used for stability and reducing conservatism of the tube. The resulting robust formulations inherit the linearity and retain the tractability of the nominal mp-NCO-tracking design problem. In this way, the design procedure of the nominal case can be applied directly to its robust counterpart with the offline computational effort independent of the number of uncertain parameters. Backing off the constraints may lead to a modification of the size of the critical regions and possibly the number of critical regions as welleither removing or adding extra regions. The approach has been demonstrated by the case studies, by adding either additive or multiplicative uncertainty, and feasibility is guaranteed at all times by using the robust controllers.

Based on the robust control framework, a similar strategy might be applied towards the mp-NCO-tracking control for continuous-time nonlinear dynamic systems in the future. Techniques for developing approximate linear surrogate models may be applied, and the gap between the nonlinear model and linear models may be computed based on bounded set propagation techniques. For guarantee feasibility, constraint back-offs can be taken into account for the model mismatch rather than the disturbances as in the robust case, while both have the effect of deviating the system responses from the nominal predictions. Moreover, disturbances can be still added to the system dynamics and evolve as model mismatch. In such a way, the design problem should be amendable to the developed robust mp-NCO-tracking design problem with the same computational effort, in which a nonlinear counterpart formulation can be developed for the nonlinear control systems similar as in the robust case. The surrogate model can be developed based on model identification techniques with the corresponding states taken as the nominal reference, and the control switching structure is calculated based on the approximate model. To guarantee feasibility of the original nonlinear dynamic system under uncertainty, a robust formulation needs to be defined for the worst-case model mismatch between the uncertain nonlinear system and the surrogate linear model. By using the advanced set propagation techniques for the nonlinear system, reachability tubes may be computed for the nonlinear responses, in which additional state feedback input can also help shrink the tube as in the robust case for linear dynamics. A potential improvement on controlling the reachability tube for the nonlinear model compared to the case with unmeasured disturbances in the linear case might lie in that, given high fidelity model of the nonlinear dynamic, a prediction can be obtained for the difference between the linear approximation and the nonlinear response, and more information about the systems is known for developing feedback control, which is applied to the original system for rejecting the effect of model mismatch. Further work will include applying more complex bounding techniques for the uncertainty propagation of nonlinear dynamic systems. Applications to higher-dimensional problems will also be considered, in which model reductions techniques may be used to reduce the order of the dynamic systems subject to some acceptable performance loss. Another direction is to investigate a more general form of the dynamic optimization problem towards an economic objective, based on which derivation of the corresponding NCOs is performed. Afterwards, multi-parametric switching structure may be characterized based the methods as proposed in the thesis, by which mp-NCO-tracking control may be applied.

Further developments based on the classifier might be considered by building a hierarchical pipeline of models based on the neural networks. In the procedure for training, only the information of labels for critical regions is used for generating the training data, while other information is discarded. In the process of generating training data, the structure detection may be taken one step further to estimate the switching times. Thus, a two-layer structure of networks might be constructed, with the first layer a classifier and the second layer consisting of different regression models for the switching times corresponding to each critical region. The efficiency of the sampling for training the classification model might be further explored. For adaptive sample data generation, more advanced techniques might be applied to fill the complete domain of interest while minimizing the cost. Further effort may be put on estimation of high variance regions by ranking points compared to the neighbouring samples, and on choosing the best candidates for new points around the selected reference points to optimally represent the local area. In the classification problem, system parameters are assumed to be identically distributed over the complete domain, while other distribution can be considered and biased sampling can be further calculated based on the corresponding statistical distributions. Moreover, the training process might benefit from the high performance distributed computing, such that the sample data can be generated by parallel computing and parallel training can be performed for different models in which suitable scheduling in a cluster or grid might be built into the training framework.

Bibliography

[1] S. S. Bahakim and L. A. Ricardez-Sandoval, "Simultaneous design and MPC-based control for dynamic systems under uncertainty: A stochastic approach," *Computers & Chemical Engineering*, vol. 63, pp. 66–81, 2014.

[2] A. E. Bryson and Y. Ho, *Applied Optimal Control*. Wiley, New York, 1975.

[3] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. characterization of the nominal solution," *Computers & Chemical Engineering*, vol. 27, pp. 1–26, 2003.

[4] A. Cervantes and L. T. Biegler, "Large-scale dae optimization using a simultaneous nlp formulation," *AIChE Journal*, vol. 44, pp. 1038–1041, 1998.

[5] L. T. Biegler, A. M. Cervantes, and A. Wächter, "Advances in simultaneous strategies for dynamic process optimization," *Chemical Engineering Science*, vol. 57, no. 4, pp. 575–593, 2002.

[6] L. T. Biegler, "Advances in simultaneous strategies for dynamic process optimization," *An overview of simultaneous strategies for dynamic optimization: Process Intensification*, vol. 46, no. 11, p. 10431053, 2007.

[7] H. Bock and K. J Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," pp. 242–247, 01 1984.

[8] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization. part 1: theoretical aspects," *Computers & Chemical Engineering*, vol. 27, no. 2, pp. 157 – 166, 2003.

[9] D. B. Leineweber, A. Schäfer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization:

Part ii: Software aspects and applications," *Computers & Chemical Engineering*, vol. 27, no. 2, pp. 167 – 174, 2003.

[10] M. H. Wright, "The interior-point revolution in constrained optimization," in *In R.Deleone, A. Murli, P. M. Pardalos, and G. Toraldo, editors, High Performance Algorithm and Software in Nonlinear Optimization*, pp. 359–381, Kluwer Academic Publishers, 1998.

[11] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[12] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Mathematical Programming*, vol. 107, no. 3, pp. 391–408, 2006.

[13] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "Users guide for npsol(version 5.10): a fortran package for nonlinear programming," *Stanford University SQL 86-1, Stanford University, Revised*, July 1998.

[14] B. Chachuat, A. B. Singer, and P. I. Barton, "Global methods for dynamic optimization and mixed-integer dynamic optimization," *Ind. Eng. Chem. Res.*, vol. 45, pp. 8373–8392, 2006.

[15] A. Flores-Tlacuahuac, S. T. Moren, and L. T. Biegler, "Global optimization of highly nonlinear dynamic systems," *Ind. Eng. Chem. Res.*, vol. 47, no. 8, pp. 2643–2655, 2008.

[16] J. B. Rawlings and R. Amrit, "Optimizing process economic performance using model predictive control," *In: L. Magni, D.M. Raimondo, and F. Allgower (Eds.), Assessment and Future Directions of Nonlinear Model Predictive Control*, vol. 384, pp. 119–138, 2009.

[17] M. Morari, Y. Arkun, and G. Stephanopoulos, "Studies in the synthesis of control structures for chemical processes. part i: Formulation of the problem. process decomposition and the classification of the control tasks. analysis of the optimizing control structures," *AIChE Journal*, vol. 26, no. 2, pp. 220–232, 1980.

[18] T. Backx, O. Bosgra, and W. Marquardt, "Integration of model predictive control and optimization of processes," tech. rep., RWTH, June 2000.

[19] T. Marlin, *Process Control*. New York: McGraw-Hill, 1995.

[20] W. Luyben, B. Tyreus, and M. Luyben, *Plantwide Process Control*. New York: McGraw-Hill, 1999.

[21] S. Engell, "Feedback control for optimal process operation," *J. Process Control*, vol. 17, pp. 203–219, 2007.

[22] E. Sequeira, M. Graells, and L. Puigjaner, "Real-time evolution of online optimization of continuous process," *Ind. Eng. Chem. Res.*, vol. 41, pp. 1815–1825, 2002.

[23] A. Helbig, O. Abel, and W. Marquardt, "Structural concepts for optimization based control of transient processes," *Progress in System and Control Theory*, vol. 26, pp. 295–311, 2007.

[24] J. V. Kadam and W. Marquardt, "Integration of economical optimization and control for intentionally transient process operation," *Lecture Notes in Control and Information Sciences*, vol. 358, pp. 419–434, 2007.

[25] M. Morari and J. Lee, "Model predictive control: past, present and future," *Computer and Chemical Engineering*, vol. 23, pp. 667–682, 1999.

[26] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[27] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[28] R. Findeisen, F. Allgöwer, and L. Biegler, *Assessment and Future Directions of Nonlinear Model Predictive Control*. Results of the international workshop entitled "Assesmentand Future Directions of Nonlinear Model Predictive Control (NMPC05), 2007.

[29] D. Angeli and J. B. Rawlings, "Receding horizon cost optimization and control for nonlinar plants," in *Proceedings of 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, (Bologna, Italy), 2010.

[30] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *Automatic Control*, vol. 57, pp. 1615–1626, 2012.

[31] L. Würth, J. B. Rawlings, and W. Marquardt, "Economic dynamic real-time optimization and nonlinear model predictive control on infinite hoirzons," in *International Symposium on Advanced Control of Chemical Process*, (Istanbul, Turkey), 2009.

[32] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design (5th ed.).* Nob Hill Publishing, 2015.

[33] L. Santos, P. Afonso, J. Castro, N. Oliveira, and L. Biegler, "On-line implementation of nonlinear mpc: An experimental case study," *Control Engineering Practice*, vol. 9, pp. 847–857, 2001.

[34] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss, "Output feedback stabilization of constrained systems with nonlinear predictive control," *International Journal of Robust and Nonlinear Control*, vol. 13, p. 211228, 2003.

[35] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[36] A. Grancharova, T. Johansen, and P. Tøndel, "Computational aspects of approximate explicit nonlinear model predictive control," in *R. Findeisen, F. Allgower and L. Biegler, eds. Assessment and Future Directions of Nonlinear Model Predictive Control*, pp. 181–192, Springer, 2007.

[37] E. N. Pistikopoulos, "Perspectives in multiparametric programming and explicit model predictive control," *AIChE Journal*, vol. 55, pp. 1918–1925, 2009.

[38] B. Srinivasan and D. Bonvin, "Dynamic optimization under uncertainty via NCO tracking: A solution model approach," *BatchPro Symposium*, vol. Plenary talk, pp. 17–35, 2004.

[39] M. Schlegel and W. Marquardt, "Detection and exploitation of the control switching structure in the solution of dynamic optimization problems," *Journal of Process Control*, vol. 16, pp. 275–290, 2006.

[40] J. V. Kadam, M. Schlegel, B. Srinivasan, D. Bonvin, and W. Marquardt, "Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation," *Journal of Process Control*, vol. 17, pp. 389–398, 2007.

[41] B. Srinivasan and D. Bonvin, "Real-time optimization of batch processes by tracking the necessary conditions of optimality," *Industrial & Engineering Chemistry Research*, vol. 46, pp. 492–504, 2007.

[42] E. N. Pistikopoulos, "From multi-parametric programming theory to mpc-on-a-chip multi-scale systems applications," *Computers & Chemical Engineering*, vol. 47, pp. 57–66, 2012.

[43] E. N. Pistikopoulos, M. C. Georgiadis, and V. Dua, *Multiparametric programming: Theory, algorithms and applications.* Weinheim: Wiley-VCH, 2007.

[44] E. N. Pistikopoulos, M. C. Georgiadis, and V. Dua, *Multi-parametric model-based control: Theory and applications.* Weinheim: Wiley-VCH, 2007.

[45] R. S. C. Lambert, *Parametric Programming Techniques for Process Engineering Problems under Uncertainty.* PhD thesis, Dept. of Chemical Engineering, Imperial College London, U.K., 2013.

[46] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines and other kernel-based learning methods.* New York, NY, USA: Cambridge University Press, 2000.

[47] Y. Tang, "Deep learning using linear support vector machines," *CoRR*, vol. abs/1306.0239, 2013.

[48] D. Bonvin and B. Srinivasan, "On the role of the necessary conditions of optimality in structuring dynamic real-time optimization schemes," *Computers & Chemical Engineering*, vol. 51, pp. 172–180, 2013.

[49] V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos, "Explicit solutions to optimal control problems for constrained continuous-time linear systems," *IEE Proceedings: Control Theory and Applications*, vol. 152, no. 4, pp. 443–452, 2005.

[50] G. François, B. Srinivasan, and D. Bonvin, "Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty," *Journal of Process Control*, vol. 15, pp. 701–712, 2005.

[51] E. N. Pistikopoulos, L. Dominguez, C. Panos, K. Kouramas, and A. Chinchuluun, "Theoretical and algorithmic advances in multi-parametric programming and control," *Computational Management Science*, vol. 9, no. 2, pp. 183–203, 2012.

[52] E. N. Pistikopoulos, N. A. Diangelakis, R. Oberdieck, M. M. Papathanasiou, I. Nascu, and M. Sun, "PAROC - an integrated framework and software platform for the optimisation and advanced model-based control of process systems," *Chemical Engineering Science*, vol. 136, pp. 115–138, 2015.

[53] L. F. Dominguez and E. N. Pistikopoulos, "Recent advances in explicit multi-parametric nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 50, pp. 609–619, 2011.

[54] H. J. Pesch, "Real-time computation of feedback controls for constrained optimal control problems part 1: Neighboring extremals," *Optimal Control Applications & Methods*, vol. 10, no. 2, pp. 130–145, 1989.

[55] K. Malanowski and H. Maurer, "Sensitivity analysis for parametric control problems with control-state constraints," *Computational Optimization & Applications*, vol. 5, pp. 254–283, 1996.

[56] K. Malanowski and H. Maurer, "Sensitivity analysis for state constrained optimal control problems," *Discrete & Continuous Dynamical Systems*, vol. 4, no. 2, pp. 241–272, 1998.

[57] K. Malanowski and H. Maurer, "Sensitivity analysis for optimal control problems subject to higher order state constraints," *Annals of Operations Research*, vol. 101, pp. 43–73, 2001.

[58] H. Maurer and H. J. Pesch, "Solution differentiability for parametric nonlinear control problems with control-state constraints," *Journal of Optimization Theory & Applications*, vol. 86, no. 2, pp. 285–309, 1995.

[59] D. Augustin and H. Maurer, "Computational sensitivity analysis for state constrained optimal control problems," *Annals of Operations Research*, vol. 101, pp. 75–99, 2001.

[60] E. Kreindler, "Additional necessary conditions for optimal control with state-variable constraints," *Journal of Optimization Theory & Applications*, vol. 38, no. 2, p. 241, 1982.

[61] R. F. Hartl, S. P. Sethi, and R. Vickson, "A survey of the maximum principle for optimal control problems with state constraints," *SIAM Review*, vol. 37, pp. 181–218, 1995.

[62] V. Dua, *Parametric Programming Techniques for Process Engineering Problems under Uncertainty*. PhD thesis, Dept. of Chemical Engineering, Imperial College London, U.K., 2000.

[63] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, "Dynamic optimization of batch processes: II. role of measurements in handling uncertainty," *Computers & Chemical Engineering*, vol. 44, pp. 27–44, 2003.

[64] L. Pontryagin, V. Boltyanskiy, R. Gamkrelidze, and Y. Mishchenko, *The Mathematical Theory of Optimal Processes*. New York, NY, USA: Wiley-Interscience, 1962.

[65] S. Gros, B. Srinivasan, and D. Bonvin, "Optimizing control based on output feedback," *Computer & Chemical Engineering*, vol. 33, pp. 191–198, 2009.

[66] S. Gros, B. Srinivasan, B. Chachuat, and D. Bonvin, "Neighboring-extremal control for singular dynamic optimization problems. I – Single-input systems," *International Journal of Control*, vol. 82, no. 6, pp. 1099–1112, 2009.

[67] S. Gros, B. Srinivasan, B. Chachuat, and D. Bonvin, "Neighboring-extremal control for singular dynamic optimization problems. II – Multiple-input systems," *International Journal of Control*, vol. 82, no. 7, pp. 1193–1211, 2009.

[68] K. B. Ariyur and M. Krstic, *Real-Time Optimization by Extremum-Seeking Control*. Wiley, 2003.

[69] D. Dochain, M. Perrier, and M. Guay, "Extremum seeking control and its application to process and reaction systems: A survey," *Mathematics & Computers in Simulation*, vol. 82, p. 369380, 2011.

[70] L. Würth, R. Hannemann, and W. Marquardt, "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization," *J. Process Control*, vol. 19, pp. 1277–1288, 2009.

[71] M. Podmajersky and M. Fikar, "Online neighboring extremal controller design for set-point-transition in presence of uncertainty," in *Proceedings of the 17th International Conference on Process Control*, (Štrbské Pleso, Slovakia), June 9-12, 2009.

[72] S. Gros, B. Chachuat, and D. Bonvin, "NCO tracking for singular control problems using neighboring extremals," in *Proceedings of the 17th IFAC World Congress*, (Seoul, South Korea), pp. 1922–1927, 2008.

[73] J. Jaschke and S. Skogestad, "NCO tracking and self-optimizing control in the context of real-time optimization," *Journal of Process Control*, vol. 21, pp. 1407–1416, 2011.

[74] A. G. Marchetti and D. Zumoffen, "On the links between real-time optimization, neighboring-extremal control and self-optimizing control," in *Control Conference (ECC), 2013 European* (E. C. C. (ECC), ed.), (Zrich, Switzerland), 2013.

[75] S. A. Deshpande, D. Bonvin, and B. Chachuat, "Directional input adaptation in parametric optimal control problems," *SIAM Journal on Control and Optimization*, vol. 50, no. 4, p. 19952024, 2012.

[76] C. Gentric, F. Pla, M. Latifi, and J. Corriou, "Optimization and nonlinear control of a batch emulsion polymerization," *Chem. Eng. J.*, vol. 75, no. 1, pp. 31–46, 1999.

[77] D. Bonvin, B. Srinivasan, and D. Hunkeler, "Control and optimization of batch processes," *IEEE Control Systems Magazine*, vol. 26, no. 6, pp. 34–45, 2006.

[78] P. Brunovský, "Regular synthesis for the linear-quadratic optimal control problem with linear control constraints," *Journal of Differential Equations*, vol. 38, no. 3, pp. 344–360, 1980.

[79] C. Bruni and D. Iacoviello, "A study on abnormality in variational and optimal control problems," *Journal of Optimization Theory & Applications*, vol. 121, pp. 41–64, 2004.

[80] B. Chachuat, B. Houska, R. Paulen, N. Peric, J. Rajyaguru, and M. E. Villanueva, "Set-theoretic approaches in analysis, estimation and control of nonlinear systems," *9th IFAC Symposium on Advanced Control of Chemical Processes*, vol. 48, no. 1, pp. 981–995, 2015.

[81] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Processes*. MOS-SIAM Series on Optimization, 2010.

[82] M. Schlegel and W. Marquardt, "Adaptive switching structure detection for the solution of dynamic optimization problems," *Industrial & Engineering Chemistry Research*, vol. 45, no. 24, pp. 8083–8094, 2006.

[83] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*. SIAM Computational Science & Engineering, 2017.

[84] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM Advances in Design and Control, 2005.

[85] M. Hovd and S. Skogestad, "Procedure for regularity control structure selection with application to the fcc process," *AIChE Journal*, vol. 39, no. 12, pp. 1938–1953, 1993.

[86] J. Spjøtvold, P. Tøndel, and T. Johansen, "A method for obtaining continuous solutions to multiparametric linear programs," in *Proceedings of the 2005 IFAC World congress*, pp. 902–908, 2005.

[87] A. Bemporad, "A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2892–2903, 2015.

[88] C. Feller, T. Johansen, and S. Olaru, "An improved algorithm for combinatorial multi-parametric quadratic programming," *Automatica*, vol. 49, no. 5, pp. 1370–1376, 2013.

[89] A. Gupta, S. Bhartiya, and P. Nataraj, "A novel approach to multiparametric quadratic programming," *Automatica*, vol. 47, no. 9, pp. 2112–2117, 2011.

[90] M. Herceg, C. Jones, M. Kvasnica, and M. Morari, "Enumeration-based approach to solving parametric linear complementarity problems," *Automatica*, vol. 62, pp. 243–248, 2015.

[91] P. Ahmadi-Moshkenani, S. Olaru, and T. Johansen, "Further results on the exploration of combinatorial tree in multi-parametric quadratic programming," in *Proceedings of the 2016 European Control Conference*, pp. 116–122, 2016.

[92] R. Oberdieck, N. Diangelakis, and E. Pistikopoulos, "Explicit model predictive control: A connected-graph approach," *Automatica*, vol. 76, pp. 103–112, 2017.

[93] M. Aly, "Survey on multiclass classification methods," *Technical Report, Caltech*, 2005.

[94] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[95] H. Haimovich, M. Seron, G. Goodwin, and J. Agüero, "A neural approximation to the explicit solution of constrained linear mpc," in *Proceedings of the 2003 European Control Conference, Cambridge, UK*, pp. 1081–1086, 2003.

[96] M. Sun, B. Chachuat, and E. Pistikopoulos, "Design of multi-parametric NCO tracking controllers for linear dyanmic systems," *Computers & cheminal engineering*, vol. 92, pp. 64–77, 2016.

[97] K. Hornik, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015.

[99] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[100] S. Ioffe and C. Szegedyn, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[101] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *CoRR*, vol. abs/1206.5533, 2012.

[102] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[103] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proceedings of 7th European Symposium on Artificial Neural Networks*, (Belgium), pp. 219–224, April 1999.

[104] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[105] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neuf11 Computtio*, vol. 7, no. 2, pp. 219–269, 1995.

[106] D. Montgomery, *Design and analysis of experiments, 5th ed.* New York, NY, USA: Wiley, 2000.

[107] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: a practical guide.* New York, NY, USA: Wiley, 2008.

[108] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A novel hybrid sequential design strategy for global surrogate modeling of computer experiments," *SIAM Journal on Scientific Computing*, vol. 33, pp. 1948–1974, 2011.

[109] D. Deschrijver, K. Crombecq, H. M. Nguyen, and T. Dhaene, "Adaptive sampling algorithm for macromodeling of parameterized s -parameter responses," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, pp. 39–45, 2011.

[110] D. Gorissen, L. D. Tommasi, K. Crombecq, and T. Dhaene, "Sequential modeling of a low noise amplifier with neural networks and active learning," *Neural Computer & applications*, vol. 18, pp. 485–494, 2009.

[111] J. Matoušek, "On the l2-discrepancy for anchored boxes," *J. Complex.*, vol. 14, pp. 527–556, Dec. 1998.

[112] L. Pronzato and W. G. Müller, "Design of computer experiments: space filling and beyond," *Statistics and Computing*, vol. 22, pp. 681–701, May 2012.

[113] D. Gorissen, I. Couckuyt, P. Demeester, and T. Dhaene, "A surrogate modeling and adaptive sampling toolbox for computer based design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, 2010.

[114] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surface of multilayer networks," *CoRR*, vol. abs/1412.0233, 2014.

[115] L. Bottou, "Stochastic gradient tricks," in *Neural Networks, Tricks of the Trade, Reloaded* (G. Montavon, G. B. Orr, and K.-R. Müller, eds.), Lecture Notes in Computer Science (LNCS 7700), pp. 430–445, Springer, 2012.

[116] T. Schaul, I. Antonoglou, and D. Silver, "Unit tests for stochastic optimization," *CoRR*, vol. abs/1312.6055, 2013.

[117] J. Duchi, J. B. Edu, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[118] I. Sutskever, *Training recurrent neural networks.* PhD thesis, Department of Computer Science, University of Toronto, Canada, 2012.

[119] A. G. Baydin, B. A. Pearlmutter, and A. A. Radul, "Automatic differentiation in machine learning: a survey," *CoRR*, vol. abs/1502.05767, 2015.

[120] Y. a. LeCun, L. Bottou, G. B. Orr, and K. R. Muller, "Efficient backprop," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700 LECTU, pp. 9–48, 2012.

[121] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[122] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga,

S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[123] P. Mhaskar, J. Liu, and P. D. Christofides, *Fault-Tolerant Process Control*. Springer, 2013.

[124] A. Bemporad and M. Morari, "Robust model predictive control: A survey," *Lecture Notes in Control and Information Sciences*, vol. 245, pp. 207–226, 1999.

[125] R. Smith, "Robust model predictive control of constrained linear systems," *Proceedings of the 2004 American Control Conference*, pp. 245–250, 2004.

[126] E. Kerrigan, *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, U.K., 2000.

[127] E. Kerrigan and J. Maciejowski, "Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution," *International Journal on Robust and Nonlinear Control*, vol. 14, pp. 395–413, 2004.

[128] W. Langson, S. Raković, I. Chryssochoos, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[129] D. Mayne, M. Seron, and S. Raković, "Robust model predictive control of constrained linear system with bounded disturbances," *Automatica*, vol. 41, pp. 1219–224, 2005.

[130] D. Mayne, S. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, pp. 1217–1222, 2006.

[131] S. Raković and K. Kouramas, "The minimal robust positively invariant set for linear discrete time systems: Approximation methods and control applications," *Proceedings of the 45th Conference on Decision and Control*, 2006.

[132] S. Raković and M. Fiacchini, "Approximate reachability analysis for linear discrete-time systems using homothety and invariance," *Proceedings of the 17th World Congress on Automatic Control, July 6-11, Seoul, Korea*, 2008.

[133] S. Raković, Kouvaritakis, M. Basil, Cannon, and C. Panos, "Fully parameterized tube model predictive control," *Robust and Nonlinear Control*, 2012.

[134] K. Kouramas, C. Panos, Faísca, and E. Pistikopoulos, "An algorithm for robust explicit/multi-parametric model predictive control," *Automatica*, vol. 49, pp. 381–398, 2013.

[135] M. Sun, M. Villanueva, E. Pistikopoulos, and B. Chachuat, "Robust multi-parametric control of continuous-time linear dynamic systems," vol. 50, pp. 4660–4665, 2017.

[136] B. Houska, F. Logist, J. V. Impe, and M. Diehl, "Robust optimization of nonlinear dynamic systems with application to a jacketed tubular reactor," *Journal of Process Control*, vol. 22, no. 6, pp. 1152–1160, 2012.

[137] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[138] M. E. Villanueva, B. Houska, and B. Chachuat, "Unified framework for the propagation of continuous-time enclosures for parametric nonlinear odes," *Journal of Global Optimization*, vol. 62, pp. 575–613, 2015.

[139] A. Kurzhanski and I. Vályi, *Ellipsoidal Calculus for Estimation and Control*. Birkhäuser, Boston, 1997.

[140] M. E. Villanueva, J. Rajyaguru, B. Houska, and B. Chachuat, "Ellipsoidal arithmetic for multivariate systems," *Computer Aided Chemical Engineering*, vol. 37, pp. 767–772, 2015.

[141] B. Houska, A. Mohammadi, and M. Diehl, "A short note on constrained linear control systems with multiplicative ellipsoidal uncertainty," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4106–4111, 2016.