Multiplexed model predictive control

Wu, Bing Fang

2009

Wu, B. F. (2009). Multiplexed model predictive control. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/19371>

<https://doi.org/10.32657/10356/19371>

Multiplexed Model Predictive Control

WU BING FANG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2009

Acknowledgements

I owe my most sincere gratitude to my supervisor, Dr. Ling Keck Voon, for his insightful guidance, patience, motivation and continuous support during my PHD study.

I wish to express my warm and sincere thanks to Dr. J. M. Maciejowski from Cambridge University for the discussions on the idea of Multiplexed Model Predictive Control, for the insightful comments and encouragement.

I also owe my gratitude to Prof. Ho Weng Khuen from National University of Singapore for the discussions on the wafer baking experiment and the constructive comments.

In addition, I would like to thank Andreas, Yan Han, Feng Yong and Andy for the valuable discussions, cooperation and support in the experimental implementations.

And last, but definitely not least, my family and friends. I could never have achieved this goal without their endless support. Thank my parents for looking after my kids when we need them. Special thanks to my husband for great patience along the way. Thanks also goes to my kids, you are the sunshine in my life. Thank my friends for sharing my joys and sorrows.

Summary

During the last two decades, Model Predictive Control (MPC) has established itself as an important form of advanced control due to its ability to deal with constraints. This results in demanding on-line optimization, hence computing resource can become an issue when applying MPC to complex systems with many inputs or with fast response times. In this thesis, a novel algorithm, called the Multiplexed MPC is proposed. The Multiplexed MPC scheme divides the MPC problem into a sequence of smaller optimizations, solves each subsystem sequentially and updates subsystem controls as soon as the solution is available, thus distributing the control moves over a complete update cycle while the total number of control moves in a given period remains the same as that of the original MPC problem. This results in reduced computational complexity and thus shorter computational time. *This reduction in computational complexity allows Multiplexed MPC to be executed at higher sampling rate, which in turn reacts faster to disturbances and thus can lead to improved performance in some cases, despite finding sub-optimal solutions to the original problem.* Stability of the Multiplexed MPC is established in this thesis. The stability proof can be easily extended to other Multiplexed MPC schemes. It is also shown that when the active constraints set is empty or known, Multiplexed MPC can be interpreted as a linear periodic state feedback law in an augmented state space. This provides a framework for systematic analysis and investigation of the properties of Multiplexed MPC. Finally, the proposed Multiplexed MPC scheme is successfully applied to a multi-zone heating process typically encountered in semiconductor manufacturing. The experimental results shows that Multiplexed MPC indeed is computationally more scalable than conventional MPC. This reduction in computational load has been gainfully used to increase the sampling rate of MMPC, leading to better disturbance rejection performance. This result is important for semiconductor wafer baking process as it could lead to increased throughput as well as reduction in wafer-to-wafer non-repeatability which will affect critical dimension of the wafer.

Table of Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Major Contributions of the Thesis	3
1.3 Related Work	5
1.4 Organization of the Thesis	7
2 Multiplexed Model Predictive Control	9
2.1 The Basic Idea	9
2.2 Problem Formulation	12
2.2.1 Preliminary	12
2.2.2 The MMPC Algorithm	14
2.2.3 Computational Complexity of MMPC	16
2.3 Stability of MMPC	16

TABLE OF CONTENTS

iv

2.3.1	Unconstrained LQ Optimal Control for Periodic Systems . .	17
2.3.2	Stability of MMPC	18
2.4	An Illustrative Example	20
2.5	Conclusion	24
3	Analysis of Multiplexed MPC	31
3.1	Unconstrained MMPC as Periodic State Feedback	32
3.2	Closed-loop Monodromy Matrix of MMPC	36
3.3	A Formula for the MMPC Cost	40
3.4	An Illustrative Example	46
3.5	Conclusion	50
4	Multiplexed MPC for Wafer Baking Process	55
4.1	The Lithography Sequence	56
4.2	Physical Modeling	57
4.3	Parameter Estimation of the Three-zone Bake Plate	64
4.3.1	Experimental Setup	64
4.3.2	Parameter Estimation	64
4.3.3	Model Validation	67
4.4	Experimental Results	68
4.5	Computational Scalability of MMPC	70
4.6	Sampling Rate and Disturbance Rejection Performance	71
4.7	Conclusion	72

TABLE OF CONTENTS

v

5 Conclusion and Future Work	82
5.1 Conclusion	82
5.2 Recommendations for Future work	84
Author's Publications	86
Bibliography	88
Appendices	95
A Derivation of the Stabilizing Terminal Weight for Linear Periodic Control Systems	96
B Derivation of $\mathcal{X}_I(K_{\sigma(k)})$	100

List of Figures

1.1	A conceptual picture of MPC.	2
2.1	Patterns of input moves for conventional ‘synchronised’ MPC (upper), and for the Multiplexed MPC (below) introduced in this thesis.	11
2.2	Disturbance at 5.1s, no constraint.	25
2.3	Disturbance at 5.1s, with constraint.	26
2.4	Disturbance at 5.6s, no constraint.	27
2.5	Disturbance at 5.6s, with constraint.	28
3.1	Step input disturbance rejection with increasing N_u , $\{u_2, u_1\}$	52
3.2	Step input disturbance rejection with increasing N_u , $\{u_1, u_2\}$	52
3.3	Effects of updating sequence, step input disturbance at 5.5s, $N_u = 5$, solid: $\{u_1, u_2\}$, dashed: $\{u_2, u_1\}$	54
4.1	The lithography sequence, extracted from [1]	56
4.2	(a) Side view of bake plate; (b) Top view of bake plate	58
4.3	Photograph of the bake plate.	62
4.4	Experimental setup diagram	65
4.5	Photograph of the experimental setup.	65

4.6	Model verification using step response. from top to bottom, step input applied at zone-1, zone-2 and zone-3.	68
4.7	Disturbance rejection performance comparison of MMPC and SMPC (Experimental result without constraints).	74
4.8	Disturbance rejection performance comparison of MMPC and SMPC (Experimental result with constraints).	75
4.9	Pattern of input update for conventional synchronized MPC('bo') and for the Multiplexed MPC('r*')).	76
4.10	Performance and computational time of MMPC and SMPC with $N_u = 5$	77
4.11	Performance and computational time of MMPC and SMPC with $N_u = 20$	78
4.12	Performance and computational time of MMPC and SMPC with $N_u = 25$	79
4.13	Plot of SMPC and MMPC maximum computational time vs control horizon.	80
4.14	Temperature responses of MMPC with sampling interval 0.4s (Solid Red) and 1.0s (Dashed-dot Red) superimposed with simulation (Dashed Blue)	81

List of Tables

2.1	Cost comparison with increasing N_u , input disturbance at 5.1s, no constraints	29
2.2	Cost comparison with increasing N_u , input disturbance at 5.1s, with constraints	29
2.3	Cost comparison with increasing N_u , input disturbance at 5.6s, no constraints	30
2.4	Cost comparison with increasing N_u , input disturbance at 5.6s, with constraints	30
3.1	Closed-loop poles for MMPC controlled plant with increasing N_u . .	40
3.2	MMPC: Eigenvalue of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ with input disturbance at 5.0s	51
3.3	MMPC: Eigenvalue of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ with input disturbance at 5.5s	53
3.4	Cost Comparison of MMPC with increasing N_u , updating sequence $\{u_2, u_1\}$	53
3.5	Cost Comparison of MMPC with increasing N_u , updating sequence $\{u_1, u_2\}$	53
3.6	MMPC: Eigenvalue of $(\hat{P}_1 - \hat{P}_2)$ with different N_u	54

LIST OF TABLES

ix

4.1	The material characteristic	63
4.2	Maximum Computational Time of MMPC and SMPC with Varying Horizon	70
4.3	Experimental and simulation results for different sampling period .	72

Chapter 1

Introduction

1.1 Motivation

Model Predictive Control (MPC), also known as receding horizon control [2, 3], has become an established control technology owing to its advantageous feature in solving problems with physical constraints as well as problems being multivariable in nature [4, 5, 6, 7, 8, 9]. Figure 1.1 presents a conceptual picture of MPC. The objective of MPC is to determine the optimal input sequence over the control horizon N_u which minimizes a suitably chosen cost function subject to a set of constraints. The cost function usually includes two components: one is the tracking error between the predicted process output $\hat{y}_{k+i|k}$ and the reference trajectory (usually known as set points) over the prediction horizon N_p , the other component is the incremental control moves over the control horizon. If the cost is quadratic, the model and the constraints are linear, the optimization problem reduces to a quadratic programming (QP) problem. At each time step, the current state of the system is measured or estimated from past measurements and inputs, then the QP problem is solved to determine an optimal future input trajectory based on the measured or estimated state. Only the first input move is injected into the plant. At the subsequent time step, the process is repeated based on the updated state

information.

Figure 1.1: A conceptual picture of MPC.

Solving such constrained optimization problem at each time step leads to high on-line MPC computation. From the algorithm complexity point of view, the QP algorithm is $O((mN_u)^3)$ [10], where m is the number of control inputs and N_u is the control horizon. As the number of decision variables ($m \times N_u$) increases, the computation complexity increases non-linearly and it may become impractical to obtain a feasible solution within the prescribed sampling time. Thus, most practical applications of MPC are restricted to chemical process control industry where the sampling rate is slow and computational resources are abundant. However, the advent of faster computers and the development of more efficient optimization algorithms, see, e.g., [11, 12], have led to applications of MPC to processes having faster dynamics. Some recent examples include MPC applications on ships [13, 14], aerospace [15, 16, 17], road vehicles [18, 19] and notably in micro scale devices [20]. A typical example of micro scale device is MEMS (Micro Electro

Mechanical Systems), which have enjoyed substantial growth in the past decade. Recent studies have estimated the market for intelligent micro-machine based systems to be around 100 Billion/year [20].

With the development of MEMS technology, it is increasingly feasible to miniaturize machines which traditionally occupy a large volume into chip scale level. There will be increasing demand for sophisticated control systems, for example MPC, to be embedded into these micro machines to achieve truly integrated “System-on-a-Chip” (SoC) or even “factory on a chip” devices. To extend MPC application to these micro-scale or embedded systems, where computation resources are limited, the ability to generate a feasible solution within the prescribed sampling period is important for a successful implementation.

At the other end of the scale is wafer production, the trend is towards multi-zone thermal processing in the semiconductor manufacturing. For example, the state of the art thermal processing system for wafer manufacturing has as many as 49 heating zones, making the system a high dimensional multivariable system [1]. Thus, a MPC algorithm which has reduced computational burden for embedded application and is scalable with respect to systems with many inputs would be very useful.

1.2 Major Contributions of the Thesis

In this thesis, a novel MPC scheme, called Multiplexed MPC (MMPC), which is scalable and has reduced computational complexity and is suitable for high speed embedded applications is proposed.

The basic idea of MMPC is to divide the original MPC problem into a sequence of optimization, solves each subsystem sequentially and updates subsystem controls as soon as the solution is available, thus distributing the control moves over a complete update cycle while the total number of control moves in a given period

remains the same. In other words, instead of solving one large problem with m inputs with computational complexity of $O((mN_u)^3)$, with multiplexed MPC, one needs only to solve m problems with one input with computational complexity of $O((N_u)^3)$, a much simpler problem especially when m is large. The resulting computational simplification allows MPC to be executed on limited capacity embedded hardware or at higher sampling rate. Executing MPC at higher sampling rate results in MPC reacting faster to disturbances and thus can lead to improved performance in some cases, despite finding sub-optimal solutions to the original problem. Figure 4.7 demonstrates the potential improvement MMPC can have over SMPC in rejecting disturbances typically encountered in the Post-Exposure Bake (PEB) step in the lithography sequence in transferring microscopic patterns from the photo-mask to semiconductor wafers (see Chapter 4 for details).

In summary, the major contributions of this thesis are:

- Proposed a novel control algorithm called Multiplexed MPC which has a lower computation burden than standard MPC.
- Established the nominal stability of the proposed Multiplexed MPC. The technique of proof is applicable to many other MMPC-like schemes.
- Developed formula for computing the quadratic cost of MMPC. The formula is useful in evaluating the effect of MMPC tuning parameters and its updating patterns.
- Applied the proposed MMPC to a real application in the semiconductor manufacturing process. The experimental results demonstrated that MMPC achieved a shorter recovery time than conventional MPC due to higher sampling rate made possible through its reduced computational complexity.

1.3 Related Work

The idea of MPC is well-known and in recent years many excellent textbooks have been written, giving a systematic and comprehensive introduction and review on this subject [3, 7, 9, 21, 22]. In addition, many survey papers, notably [4, 6, 23, 24, 25, 26] have been published: The most recent survey of industrial applications of MPC is provided in [6], stability and robustness for constrained MPC are well-addressed by [4, 23, 24]. In addition, [25] presents a review of the explicit approaches to constrained MPC and [26] surveys the literature on MPC in distributed settings. Commercial and academic tools for MPC design are also available, for example, the MPC toolbox in MATLAB [27], Multi-Parametric Toolbox (MPT) from the Automatic Control Laboratory of ETH Zurich [28] and LabVIEW from NI (National Instrument) also provides MPC functionality in its Control Design and Simulation Module [29].

The idea of MMPC is related to distributed MPC [30, 31, 32] or decentralized MPC [33, 34, 35], both dividing the optimization into smaller sub-problems. Several works have been published that propose ‘decentralized MPC’ in the sense that subsets of control inputs are updated by means of an MPC algorithm. But these usually assume that several sets of such computations are performed in parallel, on the basis of local measurements only, and that all the control inputs are then updated simultaneously. In some applications, such as formation flying of unmanned vehicles, it is assumed that the state vectors of subunits (vehicles) are distinct, and that coupling between subunits occurs only through constraints and performance measures. In [35] five different MPC-based schemes are proposed, of which four are decentralized MPC schemes of some kind. Their schemes 4 and 5 are the closest to our multiplexed scheme. In these schemes an MPC solution is solved iteratively for each control input, but it is assumed that no new sensor information arrives during the iteration, and that all the control inputs are updated simultaneously when the iterations have been completed.

In [36] a similar scheme to ours is proposed. But it is assumed that a limitation occurs on network bandwidth rather than on central computing resources. Hence optimal trajectories are computed for all the plant inputs, but these are communicated to the plant one input (or one group of inputs) at a time — with the optimization taking this communication restriction into account. If the communication sequence is fixed and periodic then the scheme proposed in [36] is essentially the same as a version of MMPC to which we previously referred as ‘scheme 1’, except that we allowed constraints on inputs and states [37]. [36] also considers the case that a feedback law is fixed for each input (or group of inputs), that the inputs are updated according to some periodic scheme, and that a heuristic is used to determine (online) the best point in the period for a given state which is not really MPC any longer, since the feedback law is assumed to be predetermined. We emphasize that the driving factor behind the development of MMPC is the operation in a processor-limited environment, motivating decomposition of the optimization to reduce computational delays. Therefore, we have not considered the impact of communication limits. Indeed, in many anticipated applications of MMPC, the communication may take place serially on a single processor, and thus communication is not a concern.

The key feature of the scheme proposed in this thesis is that the inputs are updated sequentially, and that each control update takes account of all the information available at that time, namely knowledge of all updates already performed, and of the latest sensor outputs. Readers familiar with iterative methods of solving linear equations may find the following comparison helpful: the distinction between previous proposals for distributed MPC and our proposal for MMPC is analogous to the distinction between the Jacobi and the Gauss-Seidel iterative algorithms [38]. The Jacobi algorithm updates every variable using values only from the previous iteration. The Gauss-Seidel algorithm, on the other hand, immediately uses new values of those variables that have already been updated within the current iteration. Multiplexed MPC shares this idea with the Gauss-Seidel method; the move

for the current actuator takes into account moves already made by other actuators in the same cycle of iterations.

In [39] MPC is considered with opposite assumptions to ours on update rates. There the plant inputs are considered to be updated relatively frequently, compared with the rate at which output measurements become available. It is remarked there that the predictive control law which results (with the specific assumptions made in [39]) is periodic, the period being the ratio of the input update rate to the output measurement rate (assuming this is an integer). A similar observation is central to the development in our Section 2.3.

An alternative strategy for speeding up the computations involved in MPC is off-line precomputation of the ‘pieces’ of the piecewise-affine controller which is the optimal solution [40]. But that is not feasible if the number of ‘pieces’ required is excessively large, or if the constraints or the plant model change relatively frequently.

1.4 Organization of the Thesis

The remaining chapters of the thesis will describe the Multiplexed MPC in detail.

In Chapter 2, the problem formulation of Multiplexed MPC (MMPC) is formally stated and its nominal stability is established. Illustrative examples are given to highlight the behavior of Multiplexed MPC.

Chapter 3 further studies some properties of MMPC. When the active constraints set is empty or known, MMPC can be viewed as a periodic state feedback control and can be cast in the standard periodic control framework. Connecting MMPC to periodic control enabled a systematic analysis of its closed-loop properties. Formula to calculate the quadratic cost of MMPC is obtained. The cost formula provides a useful tool in evaluating different MMPC designs. Illustrative examples are presented to show its application.

Chapter 4 applies MMPC to a wafer baking process. A brief description of the process and its model are first given. Then the experimental setup for parameter estimation and model validation are described. The experimental results showed that MMPC performed better than the conventional MPC in terms of settling time and faster rejection of disturbance.

Finally, conclusions and suggestions for future work are given in Chapter 5.

For completeness and easy reference, Appendix A presents the derivation of the stabilizing terminal weight for linear periodic control systems.

Appendix B presents the procedure of obtaining the terminal invariant set given a particular periodic controller in MMPC.

Chapter 2

Multiplexed Model Predictive Control

Model Predictive Control (MPC) has become an established control technology in the petrochemical industry, and its use is currently being pioneered in an increasingly wide range of process industries. It is also being proposed for a range of higher bandwidth applications, such as ships [13, 14], aerospace [15, 16, 17] and road vehicles [18, 19].

In this chapter, a novel control algorithm – Multiplexed MPC—is described in detail. Multiplexed MPC aims to facilitate applications of MPC where computational complexity, in particular computation time, is likely to be an issue. One can foresee that applications to embedded systems, with the MPC algorithm implemented in a chip or an FPGA [20, 41, 42, 43, 44, 45], are likely to run up against this problem.

2.1 The Basic Idea

MPC operates by solving an optimization problem on-line, in real time, to determine a plan for future operation. Only an initial portion of that plan is im-

plemented, and the process is repeated, re-planning when new information becomes available. Since numerical optimization naturally handles hard constraints, MPC offers good performance while operating close to constraint boundaries [9]. Solving a numerical optimization can be a complex problem, and for situations in which computational resource is limited, the time to find the solution can be the limiting factor in the choice of the update interval. All MPC theory to date, and as far as we know all implementations, assume that all the control inputs are updated at the same instant. Suppose that a given MPC control problem can be solved in not less than T seconds, so that the smallest possible update interval is T . The computational complexity of typical MPC problems, including time requirements, tend to vary as $O((mN_u)^3)$, where m is the number of control inputs and N_u is the control horizon. We propose to use MPC to update only one control input at a time, but to exploit the reduced complexity to update successive inputs at intervals smaller than T , typically T/m . After m updates a fresh cycle of updates begins, so that each whole cycle of updates repeats with cycle time T . We call this scheme *multiplexed MPC*, or MMPC. We assume that fresh measurements of the plant state are available at these reduced update intervals T/m . The main motivation for this scheme is the belief that in many cases the approximation involved in updating only one input at a time will be outweighed — as regards performance benefits — by the more rapid response to disturbances, which this scheme makes possible. It is often the case that “do something sooner” leads to better control than “do the optimal thing later”. Figure 2.1 shows the pattern of input moves in the Multiplexed MPC scheme with $m = 3$, compared with the conventional scheme in which the three input moves are synchronized. (We will refer to conventional MPC as *synchronized MPC* (SMPC) in the rest of this thesis.)

Various generalizations of our scheme are possible. For example, subsets of control inputs might be updated simultaneously, perhaps all the inputs in each subset being associated with one subunit. Alternatively, sensor outputs might become available one at a time (or one subset at a time), as in the common practice of

Figure 2.1: Patterns of input moves for conventional ‘synchronised’ MPC (upper), and for the Multiplexed MPC (below) introduced in this thesis.

“polling” sensors. A further generalization would be not to update each control input in a fixed sequence, but to decide in real time which input (if any) needs updating most urgently — one could call this *just-in-time MPC*.

The rest of this chapter is organized as follows. In Section 2.2, the formulation of Multiplexed MPC is presented in detail. Section 2.3 establishes the nominal stability of MMPC. Section 2.4 gives numerical simulation examples which illustrate the behavior of MMPC and compare the performance of MMPC with SMPC. Finally,

concluding remarks are given in Section 2.5.

2.2 Problem Formulation

2.2.1 Preliminary

We consider the following discrete-time linear plant model in the state-space form, with state vector $x_k \in \mathbb{R}^n$ and m scalar inputs $u_{1,k}, \dots, u_{m,k}$:

$$x_{k+1} = Ax_k + \sum_{j=1}^m B_j \Delta u_{j,k} \quad (2.1)$$

where each B_j is a column vector and $\Delta u_{j,k} = u_{j,k} - u_{j,k-1}$. (This could be generalized to the case where $B_j \in \mathbb{R}^{n \times p_j}$ and $\Delta u_{j,k} \in \mathbb{R}^{p_j}$, with $\sum_j p_j$ inputs.)

We assume that $(A, [B_1, \dots, B_m])$ is stabilizable¹. For ease of notation, when we drop the index j , we mean the complete B matrix and the input vector so that the system (2.1) may be written as

$$x_{k+1} = Ax_k + B\Delta u_k$$

The unique advantage of MPC, compared with other control strategies, is its capacity to take account of constraints in a systematic manner. As usual in MPC, we will suppose that constraints may exist on the input amplitudes, $\|u_k\|_\infty \leq U$, on the input moves, $\|\Delta u_k\|_\infty \leq D$, and on states, $Mx_k \leq v$. (These can all be generalized substantially, so long as linear inequalities are retained.)

We assume that at time step k the complete state vector x_k is known exactly

¹The assumption that $(A, [B_1, \dots, B_m])$ is stabilizable is not used directly in MMPC design. However, it is used to ensure that the plant can be stabilized by the standard MPC design. For MMPC, we have implicitly assumed that $(A, B_{\sigma(k)})$ is stabilizable. In general, $(A, [B_1, \dots, B_m])$ stabilizable does not necessarily imply that $(A, B_{\sigma(k)})$ is stabilizable or vice versa and conditions under which $(A, [B_1, \dots, B_m])$ stabilizable implies that $(A, B_{\sigma(k)})$ is stabilizable or vice versa is a subject of current research.

from measurements. We will consider only the regulation problem in detail, but tracking problems, especially those with non-zero constant references, can be easily transformed into equivalent regulation problems.

Multiplexed MPC, at discrete-time index k , changes only plant input $(k \bmod m) + 1$. (We assume, without loss of generality, that we update input 1 at the initial time.) The following indexing function identifies the input channel to be moved at each step

$$\sigma(k) = (k \bmod m) + 1 \quad (2.2)$$

so that the asynchronous nature of the multiplexed control moves, as illustrated in Fig. 2.1, is captured by the constraint

$$\Delta u_{j,k} = 0 \text{ if } j \neq \sigma(k). \quad (2.3)$$

It is then possible to rewrite the system dynamics (2.1) as a linear periodic time-varying system

$$x_{k+1} = Ax_k + B_{\sigma(k)} \Delta \tilde{u}_k \quad (2.4)$$

where $\Delta \tilde{u}_k = \Delta u_{\sigma(k),k}$. From this point onwards, we use this periodic description of the plant so that we can draw on known results for periodic time-varying systems.

Let $N = (N_u - 1)m + 1$ where N_u is the control horizon, a design parameter which will be used to denote the number of control moves to be optimized *per input channel* of the original system (2.1). The N -step prediction model at time k for the system described by (2.4) is

$$\vec{X}_{k+1|k} = \Phi x_{k|k} + G_{\sigma(k)} \Delta \vec{U}_{k|k} \quad (2.5)$$

where

$$\vec{X}_{k+1|k} = \begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix}, \quad \Delta \vec{U}_{k|k} = \begin{bmatrix} \Delta \tilde{u}_{k|k} \\ \Delta \tilde{u}_{k+1|k} \\ \vdots \\ \Delta \tilde{u}_{k+N-1|k} \end{bmatrix}, \quad \Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix},$$

$$G_{\sigma(k)} = \begin{bmatrix} B_{\sigma(k)} & 0 & \dots & 0 \\ AB_{\sigma(k)} & B_{\sigma(k+1)} & \dots & 0 \\ \vdots & & \ddots & \\ A^{N-1}B_{\sigma(k)} & \dots & AB_{\sigma(k+N-2)} & B_{\sigma(k+N-1)} \end{bmatrix}$$

$\Delta \tilde{u}_{k+i|k} = \Delta u_{\sigma(k+i), k+i|k}$ denotes the prediction made at time k of a control move to be executed at time $k+i$, and $x_{k+i|k}$ denotes the corresponding prediction of x_{k+i} , made at time k .

2.2.2 The MMPC Algorithm

Before presenting the MMPC algorithm, we introduce the following notations.

Let $\mathbb{X}$ and $\mathbb{U}_k$ be compact polyhedral sets containing the origin in their interior. These sets correspond to admissible values of states and inputs, respectively. $\mathbb{U}_k$ is allowed to depend on k so as to allow constraints on $\tilde{u}_k$ as well as on $\Delta \tilde{u}_k$. $K_{\sigma(k)}$ denotes a pre-specified stabilizing linear periodic state-feedback controller for (2.4); $\mathcal{X}_I(K_{\sigma(k)})$ denotes the set in which none of the constraints is active, and which is the maximum positively invariant set [46] for the linear periodic system (2.4) when the feedback controller $\Delta \tilde{u}_k = K_{\sigma(k)}x_k$ is applied, namely

$$x_k \in \mathcal{X}_I(K_{\sigma(k)}) \Rightarrow K_{\sigma(k)}x_k \in \mathbb{U}_k \text{ and } (A + B_{\sigma(k)}K_{\sigma(k)})x_k \in \mathcal{X}_I(K_{\sigma(k)})$$

and of course $\mathcal{X}_I(K_{\sigma(k)}) \subset \mathbb{X}$.

Some assumptions must be made about those inputs which have already been

planned but which have not yet been executed. We will assume that all such planned decisions are known to the controller, and that it assumes that they will be executed as planned, i.e.,

$$\Delta \tilde{u}_{k+i|k} = \Delta \tilde{u}_{k+i|k-1}, \quad i \neq 0, m, 2m, \dots \quad (2.6)$$

(This assumption will usually be false, because new decisions will be made in the light of new measurements.)

MMPC solves the following finite-time constrained linear periodic control problem:

$$\begin{aligned} \mathcal{P}_{\sigma(k)}(x_k) : \quad & \text{Minimise} \quad J_k = F(x_{k+N|k}) + \sum_{i=0}^{N-1} (\|x_{k+i+1|k}\|_q^2 + \|\Delta \tilde{u}_{k+i|k}\|_r^2) \\ & \text{wrt} \quad \Delta \tilde{u}_{k+i|k}, \quad (i = 0, m, 2m, \dots, N-1) \\ & \text{s.t.} \quad \Delta \tilde{u}_{k+i|k} \in \mathbb{U}_k, \quad (i = 0, \dots, N-1) \\ & \quad x_{k+i|k} \in \mathbb{X}, \quad (i = 1, \dots, N) \\ & \quad x_{k+N|k} \in \mathcal{X}_I(K_{\sigma(k)}) \\ & \quad x_{k+i+1|k} = Ax_{k+i|k} + B_{\sigma(k+i)}\Delta \tilde{u}_{k+i|k} \\ & \quad \Delta \tilde{u}_{k+i|k} = \Delta \tilde{u}_{k+i|k-1}, \quad (i \neq 0, m, 2m, \dots, N-1) \end{aligned} \quad (2.7)$$

where $q = q^T \geq 0$ and $r = r^T > 0$ are defined as the weighting matrices² for states and control increments with $q \in \mathbb{R}^{n \times n}$ and $r \in \mathbb{R}$. $F(x_{k+N|k}) \geq 0$ is a suitably chosen terminal cost. Please refer to Appendix B for the detailed procedure for deriving $\mathcal{X}_I(K_{\sigma(k)})$ given a local controller $K_{\sigma(k)}$.

We denote the resulting optimizing control sequence as $\Delta \mathbf{u}^o(x_k)$. Only the first control $\Delta \tilde{u}_k^o$ in $\Delta \mathbf{u}^o(x_k)$ is applied to the system at time k , so that we apply the predictive control in the usual receding-horizon manner.

In MMPC, there are essentially m MPC controllers, operating in sequence, in a cyclic manner. They share information, however, in the sense that the complete

²The weighting matrices are in lower case because the uppercase Q and R are reserved for the weighting matrices in Chapter 3.

plant state is available to each controller — although not at the same times — and the currently planned future moves of each controller are also available to all the others.

Remark 2.1 *From the MMPC algorithm stated above, we know that at time step k , the future trajectory of only one input of system (2.1) is optimized and the future behavior of the other inputs are assumed inherited from the previous step $k-1$. Therefore, the number of decision variables in MMPC is N_u , which has been reduced by m as compared with mN_u in traditional synchronized MPC.*

2.2.3 Computational Complexity of MMPC

The computational complexity of typical MPC problems is $O((mN_u)^3)$, where m is the number of control inputs and N_u is the control horizon. However, the MMPC scheme solves m control problems of computational complexity $O((N_u)^3)$. Thus, the total computation for MMPC is $m \times O((N_u)^3)$ and this compares favorably with standard MPC's $O((mN_u)^3)$ especially when N_u is large. As a simple illustration, consider a Two-Input-Two-Output example of $m = 2$ and $N_u = 5$. In one sampling period, SMPC needs to solve computational problem of complexity $O((10)^3)$ or $O(1000)$ while MMPC only need to solve computational problem of complexity $O((5)^3)$ or $O(125)$. Therefore, the total computation burden has been reduced from $O(1000)$ to $2 \times O(125)$. In a recent study shown in [47], the computational advantage of MMPC over SMPC has been verified through a large commercial turbofan engine application.

2.3 Stability of MMPC

Before we move on to the stability analysis of MMPC, a brief review of unconstrained LQ optimal control for periodic systems will be given as it is the basis for

stability proof in this section. Multiplexed MPC is a periodic control scheme, thus the stability criteria for periodic control system provides the basis for the stability of constrained Multiplexed MPC.

2.3.1 Unconstrained LQ Optimal Control for Periodic Systems

The following results on unconstrained infinite-time linear quadratic control of periodic systems are known [48]. Consider the periodic plant (2.4) and the quadratic cost function

$$J_k = \sum_{i=0}^{\infty} (\|x_{k+i+1}\|_q^2 + \|\Delta\tilde{u}_{k+i}\|_r^2) \quad (2.8)$$

Then this cost is minimized by finding $\bar{P}_i$, $i = 1, \dots, m$, the Symmetric, Periodic and Positive Semidefinite (SPPS) solution of the following Discrete-time Periodic Riccati Equation (DPRE)

$$P_k = A^T P_{k+1} A - A^T P_{k+1} B_{\sigma(k)} (B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A + q \quad (2.9)$$

and setting

$$\Delta\tilde{u}_k = K_{\sigma(k)} x_k \quad (2.10)$$

where

$$K_{\sigma(k)} = -(B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} A \quad (2.11)$$

Remark 2.2 The set of periodic gains $\left\{ K_j : j = 1, \dots, m \right\}$ is stabilizing, namely the *monodromy matrix* [49]

$$\Psi_1 = \Phi_m \Phi_{m-1} \dots \Phi_2 \Phi_1 \quad (2.12)$$

has all its eigenvalues inside the unit cycle, where

$$\Phi_j = A + B_j K_j \quad (2.13)$$

It is a standard fact that this is a necessary and sufficient condition for closed-loop stability of a linear periodic system, and that the eigenvalues of the monodromy matrix is invariant under cyclic permutations of the Φ_j matrices [50].

Remark 2.3 In [48], conditions are established for the existence of the optimal solution, which generalize the familiar conditions for the existence of solutions to LQ problems for LTI systems. In this thesis we shall assume that such conditions are satisfied.

2.3.2 Stability of MMPC

Theorem 2.3.1 Assume that $(A, B_{\sigma(k)})$ is stabilizable, $q = q^T \geq 0$, $r = r^T > 0$ and the system (2.1) is feasible at the initial time, the MMPC, obtained by solving the finite-time constrained linear period control problems (2.7), gives closed-loop stability if $F(x_{k+N|k})$ is the value function of the unconstrained infinite horizon periodic optimal control problem with initial condition $x_{k+N|k}$, namely if

$$F(x_{k+N|k}) = \min_{\Delta \tilde{u}_{k+i|k}} \left\{ \sum_{i=N}^{\infty} (\|x_{k+i+1|k}\|_q^2 + \|\Delta \tilde{u}_{k+i|k}\|_r^2) \right\} = x_{k+N|k}^T \bar{P}_{\sigma(k+N)} x_{k+N|k} \quad (2.14)$$

where $\bar{P}_{\sigma(k+N)}$ is defined as above.

Proof: The proof depends on the constrained optimization being feasible at each step, and the feasibility at any particular time step depends on the details of the constrained optimization problem that is being solved. For the nominal case with a perfect model and in the absence of disturbances, if feasible solutions are obtained over an initial period, then feasibility is assured thereafter.

According to [24, page 797], if the following four conditions hold:

- (1) state constraints are satisfied in the terminal constraint set,
- (2) the terminal constraint set is positively invariant under a local controller,
- (3) the local controller satisfies control constraints in the terminal constraint set,
- (4) the terminal cost is a local Lyapunov function under the same local controller.

then closed loop stability is obtained.

In our set up, conditions (1)–(3) are satisfied by assumption since $x_{k+N} \in \mathcal{X}_I(K_{\sigma(k)})$. We have only to check that condition (4) is satisfied, namely that $F(x_{k+N+1|k+1}) - F(x_{k+N|k}) \leq 0$.

That $F(x_{k+N|k})$ as defined in (2.14) is the value function of the unconstrained infinite horizon periodic optimal control problem is shown in [48]. We now proceed using a standard argument. Let $\Delta \tilde{u}_{k+i|k}^*$ denote the optimal solution to (2.7), for $i = 0, m, \dots, N-1$, and let $x_{k+i|k}^*$ denote the corresponding state sequence, for $i = 1, 2, \dots$. By Bellman's optimality principle we have that $x_{k+N+1|k+1}^* = x_{k+N+1|k}^*$, and hence we have

$$F(x_{k+N|k}) = F(x_{k+N+1|k+1}) + \|x_{k+N+1|k}^*\|_q^2 + \|\Delta \tilde{u}_{k+N|k}^*\|_r^2 \quad (2.15)$$

from which

$$F(x_{k+N+1|k+1}) - F(x_{k+N|k}) = -\|x_{k+N+1|k}^*\|_q^2 - \|\Delta \tilde{u}_{k+N|k}^*\|_r^2 \leq 0 \quad (2.16)$$

follows, with equality satisfied if and only if $x_{k+N|k}^* = \Delta \tilde{u}_{k+N|k}^* = 0$. This shows that $F(x_{k+N|k})$ is a Lyapunov function in some neighborhood of 0, in particular for $x_k \in \mathcal{X}_I(K_{\sigma(k)})$. ■

It can be seen that this result, and its proof, depend only on using an appropriate terminal cost $F(\cdot)$, and not at all on the details of the constrained optimization

over the horizon of length N . Consequently the result and its proof hold for any other MMPC scheme which involves constrained optimization over a finite horizon, providing that the terminal cost is given by (2.14). For example, for a particular 3-input system, it may be beneficial to update the inputs in the sequence $(1, 2, 1, 3, 1, 2, 1, 3, \dots)$, thus updating one of the inputs twice as often as the others. The stability of such a scheme is proved by the argument used above, providing that the sequencing function $\sigma(\cdot)$ is redefined appropriately.

2.4 An Illustrative Example

In this section, we give a numerical example to illustrate the proposed MMPC scheme. In particular, we focus on the disturbance rejection performance and compare it with the conventional “synchronized MPC” (SMPC) scheme. The simulations were carried out with the plant modeled in continuous-time and the controller modeled in discrete-time.

The plant has a continuous-time model of

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{7s+1} & \frac{1}{3s+1} \\ \frac{2}{8s+1} & \frac{1}{4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \quad (2.17)$$

Although simple, such dynamics are representative of what is commonly encountered in, e.g. multiple chamber or multi-zone temperature control problems, where the temperatures of the individual chamber/zone need to be controlled independently in the presence of thermal interactions between the chambers/zones. One approach is to perform multivariable control design, but this may lead to a complex centralized controller if the number of chambers/zones is large. An alternative is to employ decentralized control where each chamber is assigned a SISO controller suitably de-tuned to account for the inter-loop interactions. Here we propose the Multiplexed MPC approach.

The plant has two inputs and hence $m = 2$ for this example. We chose the sampling time to be $T = 1s$. More specifically, for SMPC, the states were measured at time interval of $1s$ and both u_1 and u_2 were changed at the same time and held constant over a period of $1s$. For MMPC scheme, the states were measured at $T/m = 0.5s$ with u_1 and u_2 alternatively applied at $0.5s$ intervals, but each held constant over a period of $T = 1s$, that is, u_1 is updated at times $(0, 1s, 2s, \dots)$ and u_2 is updated at times $(0.5s, 1.5s, \dots)$.

The simulation parameters are

$$q = C^T C, \quad r = I$$

with the discrete time state space model of

$$\begin{aligned} x_{k+1} &= Ax_k + B\Delta u_k \\ y_k &= Cx_k \end{aligned}$$

The terminal periodic controller gain $K_{\sigma(k)}$ are

$$\begin{aligned} K_1 &= \begin{bmatrix} 0.7289 & -2.4886 & 0.9829 & -1.9283 & 0.2203 & -1.0224 \end{bmatrix} \\ K_2 &= \begin{bmatrix} -1.1941 & 0.2884 & -1.9604 & 0.0983 & -0.9706 & -0.2091 \end{bmatrix} \end{aligned}$$

Since there is no state constraint, the admissible states set $\mathbb{X}$ is the whole state space, while the inputs admissible set $\mathbb{U}_k$ is the corresponding polyhedral which is defined by input constraints

$$\mathbb{U}_k \triangleq \{\Delta \tilde{u}_{k+i} : \tilde{u}_{min} - \tilde{u}_{k+i-1|k} \leq \Delta \tilde{u}_{k+i|k} \leq \tilde{u}_{max} - \tilde{u}_{k+i-1|k} \quad \text{for } i = 0, 1, \dots, N-1.\}$$

The terminal set $\mathcal{X}_I(K_{\sigma(k)})$ is actually the maximal output admissible set ([51]) for

the closed loop system using the local periodic controller $K_{\sigma(k)}$ and defined as

$$\mathcal{X}_I(K_{\sigma(k)}) \triangleq \{x : K_{\sigma(k)}(A + B_{\sigma(k)}K_{\sigma(k)})^i x \in \mathbb{U}_k \quad \text{for } i = 0, 1, \dots\}.$$

The terminal set $\mathcal{X}_I(K_{\sigma(k)})$ can be computed using the procedure detailed in Appendix B. For this example, it is a set of 92 inequalities.

Figure 2.2 shows the simulation result when a unit step input disturbance was introduced at time 5.1s, i.e. just after input 1 has been updated. The upper half of the figure compares the closed-loop responses between SMPC and MMPC scheme, with the design parameter $N_u = 5$. It can be seen that due to a faster reaction time, the MMPC was more effective in reducing the peak overshoot at the plant outputs, although there were larger undershoots. The lower half of the figure shows MMPC scheme for $N_u = 1, \dots, 5$. With increasing N_u the peak on the output after a disturbance is reduced but the undershoot increases.

Figure 2.3 shows the same simulation scenario when input constraints of $|u_1| \leq 1.1$ and $|u_2| \leq 1.1$ were added. It can be seen that the undershoot in MMPC scheme has been significantly reduced. Again, increasing N_u in MMPC scheme has the effect of reducing the peak overshoot at the plant outputs.

MMPC are periodic control schemes, thus their performance depends on the time at which the disturbance occurs. Figure 2.4 and 2.5 show a similar set of simulation results but when the input disturbance occurs at time 5.6s, i.e. just after input 2 has been updated. From Figure 2.5, it can be seen that MMPC still offering some advantage in the two outputs with much less peak overshoot although there was a little bit bigger undershoot in channel y_2 . Again, we notice that increasing N_u in MMPC scheme has the effect of reducing the peak overshoot at the plant outputs, and adding constraints on the inputs helps to reduce the undershoots.

Table 2.1³ shows the cost of MMPC and SMPC in Figure 2.2 by summing up the squares of tracking errors and incremental controls at every 0.1s interval as an approximation to the integrals of $\int_0^\infty e(t)^T e(t)dt$ and $\int_0^\infty \Delta u(t)^T \Delta u(t)dt$. The infinite horizon costs in the tables give an indication of the cost functions of the MMPC shown in (2.7) when $q = C^T C$, $r = I$ and $w = 0$ is the reference signal. J_m and J_s are the sum of the two integrals for the MMPC and SMPC schemes respectively. Similarly, $J_{x,m}$ and $J_{x,s}$ are the values of $\int_0^\infty e(t)^T e(t)dt$, $J_{u,m}$ and $J_{u,s}$ are the values of $\int_0^\infty \Delta u(t)^T \Delta u(t)dt$ for the MMPC and SMPC schemes respectively. It could be observed that as N_u increases J_m also increases. A further investigation shows that $J_{x,m}$ decreases with N_u increasing from 1 to 3 and then increases again as N_u increases from 3 to 5. For SMPC, J_s , $J_{x,s}$ and $J_{u,s}$ remain unchanged as N_u increases, as in this case the control law of SMPC is a fixed linear feedback since there is no constraint. $J_{x,s}$ is apparently bigger than $J_{x,m}$, which is consistent with the simulation result shown in Figure 2.2.

Table 2.2 presents the values of the integral of $\int_0^\infty e(t)^T e(t)dt$ and $\int_0^\infty \Delta u(t)^T \Delta u(t)dt$ for the SMPC and MMPC schemes when the input constraints becomes active and the input disturbance enters at 5.1s. As could be seen, since the undershoot in MMPC has been greatly reduced as N_u increases, with approximately same overshoots for different N_u , increasing N_u leads to smaller tracking error as could be observed from the trend of $J_{x,m}$. Since the peak overshoots of the outputs in SMPC are apparently bigger than those in MMPC, the tracking error of MMPC is surely smaller than that of SMPC. Note that with constraints, the control law of SMPC is piecewise linear, thus J_s , $J_{x,s}$ and $J_{u,s}$ will be different with varying N_u .

Table 2.3 and 2.4 list the values of the integral of $\int_0^\infty e(t)^T e(t)dt$ and $\int_0^\infty \Delta u(t)^T \Delta u(t)dt$ for the SMPC and MMPC schemes when disturbance occurred at 5.6s. Trend sim-

³Tables 2.1-2.4 give a quantitative measure of the closed-loop performance in the form of sum of the squares of tracking errors and the control increments. From the comparison listed in Tables 2.1-2.4, we observed that J_m is not always smaller than J_s . Since MMPC is a periodic control scheme, the time instant when disturbance enters the system will have an impact on the performance of MMPC. These observations motivated us to find a way to compute the analytical cost of MMPC, which led to Chapter 3.

ilar to those in Table 2.1 and 2.2 could be observed. With constraints, the tracking error decreases as N_u increases. In either case, $J_{x,m}$ is smaller than $J_{x,s}$ as could be observed from the simulation results.

In this example MMPC gives better performance than SMPC, as judged by recovery from step disturbances. This cannot be expected to be true always. But even if the performance is worse than that of SMPC, MMPC scheme offers considerably lower computational complexity, which might make it the more desirable control algorithm.

2.5 Conclusion

In this chapter, a novel control scheme known as *Multiplexed* MPC was proposed, which is expected to be of practical benefit because it offers reduced computational complexity. The proposed MMPC scheme has been proved to be nominally stable. The nominal stability of a large class of other Multiplexed MPC schemes follows by the same argument as we used in this chapter.

It is interesting, and potentially important, to observe that the assumption of equal intervals between the updates of plant inputs is not essential to our proposal. Any pattern of update intervals can be supported, providing that it repeats in subsequent update cycles.

The computation complexity of MMPC and SMPC are also shown and compared. An illustrative example has been included to show some performance benefit of MMPC over conventional MPC in the form of faster reactions to disturbances. The numerical cost comparison between MMPC and SMPC has been presented through comparing the sum of squares of the tracking errors and the control increments. It would be useful to obtain analytical formula to compute the cost of MMPC and this will be elaborated in the next chapter.

Figure 2.2: Disturbance at 5.1s, no constraint.

Figure 2.3: Disturbance at 5.1s, with constraint.

Figure 2.4: Disturbance at 5.6s, no constraint.

Figure 2.5: Disturbance at 5.6s, with constraint.

Table 2.1: Cost comparison with increasing N_u , input disturbance at 5.1s, no constraints

Cost	N_u				
	1	2	3	4	5
$J_m \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	1.8493	1.8787	1.9141	1.9459	1.9758
$J_{x,m} \approx \int_0^\infty e(t)^T e(t) dt$	0.4500	0.3827	0.3708	0.3838	0.4072
$J_{u,m} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.3993	1.4960	1.5433	1.5621	1.5686
$J_s \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	2.2749	2.2749	2.2749	2.2749	2.2749
$J_{x,s} \approx \int_0^\infty e(t)^T e(t) dt$	1.0844	1.0844	1.0844	1.0844	1.0844
$J_{u,s} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.1905	1.1905	1.1905	1.1905	1.1905

Table 2.2: Cost comparison with increasing N_u , input disturbance at 5.1s, with constraints

Cost	N_u				
	1	2	3	4	5
$J_m \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	1.8495	1.8864	1.9495	2.0130	2.0737
$J_{x,m} \approx \int_0^\infty e(t)^T e(t) dt$	0.5271	0.4551	0.4020	0.3701	0.3501
$J_{u,m} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.3223	1.4313	1.5475	1.6430	1.7236
$J_s \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	2.3258	2.3259	2.3250	2.3249	2.3249
$J_{x,s} \approx \int_0^\infty e(t)^T e(t) dt$	1.1858	1.2079	1.1985	1.1970	1.1970
$J_{u,s} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.1400	1.1180	1.1265	1.1279	1.1279

Table 2.3: Cost comparison with increasing N_u , input disturbance at 5.6s, no constraints

Cost	N_u				
	1	2	3	4	5
$J_m \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	1.9278	1.9759	2.0650	2.1384	2.1862
$J_{x,m} \approx \int_0^\infty e(t)^T e(t) dt$	0.4957	0.4249	0.4093	0.4182	0.4357
$J_{u,m} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.4320	1.5510	1.6557	1.7202	1.7505
$J_s \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	1.9925	1.9925	1.9925	1.9925	1.9925
$J_{x,s} \approx \int_0^\infty e(t)^T e(t) dt$	1.0191	1.0191	1.0191	1.0191	1.0191
$J_{u,s} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	0.9734	0.9734	0.9734	0.9734	0.9734

Table 2.4: Cost comparison with increasing N_u , input disturbance at 5.6s, with constraints

Cost	N_u				
	1	2	3	4	5
$J_m \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	1.9276	1.9634	2.0469	2.1484	2.2447
$J_{x,m} \approx \int_0^\infty e(t)^T e(t) dt$	0.5327	0.4603	0.4150	0.3856	0.3661
$J_{u,m} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	1.3949	1.5031	1.6319	1.7627	1.8786
$J_s \approx \int_0^\infty [e(t)^T e(t) + \Delta u(t)^T \Delta u(t)] dt$	2.0354	2.0351	2.0338	2.0336	2.0336
$J_{x,s} \approx \int_0^\infty e(t)^T e(t) dt$	1.1095	1.1198	1.1094	1.1077	1.1077
$J_{u,s} \approx \int_0^\infty \Delta u(t)^T \Delta u(t) dt$	0.9259	0.9153	0.9244	0.9259	0.9259

Chapter 3

Analysis of Multiplexed MPC

In Chapter 2, the stability of Multiplexed MPC was established and an illustrative example has been presented which compared the performance between SMPC and MMPC quantitatively by summing up the data points obtained from the mixed continuous-time and discrete-time simulations. The optimal cost obtained with MMPC, even in the case that all constraints are inactive, is not given by $x_k^T \bar{P}_{\sigma(k)} x_k$, as might be expected from (2.14), because of the restricted degrees of freedom of the optimizer at each step. Those inputs which are not being optimized, but whose future moves have already been planned, act like measured disturbances in conventional control. An alternative view is to consider them as initial conditions for the optimization. In this chapter we shall pursue that view, in order to obtain an expression for the cost obtained with *any* linear periodic stabilizing controller, which updates the plant inputs sequentially as in MMPC. As a special case, the cost of MMPC, assuming that constraints are never active, can be obtained. We will show that under this assumption, MMPC can be interpreted as a linear periodic state feedback law in an augmented state space.

Note that a similar development could be used to compute the optimal MMPC cost if the set of active constraints was constant and known, and that the nature of the MMPC control law in such circumstances is also linear periodic. Consequently

the MMPC control law in the presence of constraints is piecewise-linear-periodic; as in the standard MPC case, the “pieces” correspond to regions of the state space in which the set of active constraints remains constant.

The analysis carried out in this chapter, in particular Proposition 3.3.1, facilitates performance evaluation of MMPC in certain circumstances. For example, it is useful in evaluating the effect of design choice available in MMPC (i.e. the control horizon of N_u or updating sequence, $\{u_1, u_2, u_3, \dots\}$ or $\{u_2, u_3, u_1, \dots\}$, etc. has on closed-loop performance.

3.1 Unconstrained MMPC as Periodic State Feedback

Recall that in Chapter 2, MMPC uses the model

$$x_{k+1} = Ax_k + B_{\sigma(k)}\Delta\tilde{u}_k \quad (3.1)$$

where $\Delta\tilde{u}_k = \Delta u_{\sigma(k),k}$, and the N -step prediction model for the system at time k is developed as

$$\vec{X}_{k+1|k} = \Phi x_{k|k} + G_{\sigma(k)}\Delta\vec{U}_{k|k} \quad (3.2)$$

where

$$\vec{X}_{k+1|k} = \begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix}, \quad \Delta\vec{U}_{k|k} = \begin{bmatrix} \Delta\tilde{u}_{k|k} \\ \Delta\tilde{u}_{k+1|k} \\ \vdots \\ \Delta\tilde{u}_{k+N-1|k} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} A \\ A^2 \\ \dots \\ A^N \end{bmatrix} \quad G_{\sigma(k)} = \begin{bmatrix} B_{\sigma(k)} & 0 & \dots & 0 \\ AB_{\sigma(k)} & B_{\sigma(k+1)} & \dots & 0 \\ \vdots & & \ddots & \\ A^{N-1}B_{\sigma(k)} & \dots & AB_{\sigma(k+N-2)} & B_{\sigma(k+N-1)} \end{bmatrix}$$

and $\Delta \tilde{u}_{k+i|k}$ denotes the prediction made at time k of a control move to be executed at time $k+i$.

Recall also that $N = (N_u - 1)m + 1$ where N_u is the control horizon, a design parameter which denotes the number of control moves to be optimized per input channel of the original system (2.1). By grouping the predicted control signals into m vectors, the prediction model (3.2) can be re-written as

$$\vec{X}_{k+1|k} = \Phi x_{k|k} + g_1^{\sigma(k)} \Delta \vec{u}_{0|k} + g_2^{\sigma(k)} \Delta \vec{u}_{k,1|k} + \dots + g_m^{\sigma(k)} \Delta \vec{u}_{k,m-1|k} \quad (3.3)$$

where, for $i = 1, 2, \dots, m$,

$$\Delta \vec{u}_{k,i|k} = \begin{bmatrix} \Delta \tilde{u}_{k+i|k} \\ \Delta \tilde{u}_{k+m+i|k} \\ \vdots \\ \Delta \tilde{u}_{k+(N_u-2)m+i|k} \end{bmatrix}$$

and

$$\Delta \vec{u}_{0|k} = \begin{bmatrix} \Delta \tilde{u}_{k|k} \\ \Delta \tilde{u}_{k+m|k} \\ \vdots \\ \Delta \tilde{u}_{k+(N_u-1)m|k} \end{bmatrix} = \begin{bmatrix} \Delta \tilde{u}_{k|k} \\ \Delta \vec{u}_{k,m|k} \end{bmatrix} \quad (3.4)$$

In addition, $g_i^{\sigma(k)}$, ($i = 1, \dots, m$) are matrices whose columns are columns of the $G_{\sigma(k)}$ matrix, namely, $g_1^{\sigma(k)}$ is the matrix whose columns are columns 1, 1 +

$m, \dots, 1 + (N_u - 1)m$ of the matrix $G_{\sigma(k)}$, while $g_i^{\sigma(k)}, (i = 2, \dots, m)$ contains columns $i, i + m, \dots, i + (N_u - 2)m$ columns of $G_{\sigma(k)}$.

In MMPC only $\Delta \vec{u}_{0|k}$ is taken as the decision variable at time k , and appropriate assumptions are made about $\Delta \vec{u}_{k,i|k}, i = 1, \dots, m - 1$. Note that the length of $\Delta \vec{u}_{0|k}$ is N_u while the length of $\Delta \vec{u}_{k,i|k}$ for $i = 1, \dots, m - 1$ is $N_u - 1$. When $N_u = 1$, $\Delta \vec{u}_{k,i|k}, i = 1, \dots, m - 1$, become zero-length vectors. In MMPC we assume that the $\Delta \vec{u}_{k,i|k}$ are those inputs which have already been planned in previous steps but have not yet been executed, namely

$$\Delta \vec{u}_{k,i|k} = \Delta \vec{u}_{k-1,i+1|k-1}, \quad (i = 1, \dots, m - 1). \quad (3.5)$$

To this end, we define the vector $\Delta \vec{u}_{k|k-1}^p$ to keep track of these predicted planned, but not yet been executed control moves

$$\Delta \vec{u}_{k|k-1}^p = \begin{bmatrix} \Delta \vec{u}_{k-1,2|k-1} \\ \Delta \vec{u}_{k-1,3|k-1} \\ \vdots \\ \Delta \vec{u}_{k-1,m|k-1} \end{bmatrix}. \quad (3.6)$$

Equation (3.3) can be re-written as

$$\vec{X}_{k+1|k} = \Phi x_{k|k} + g_1^{\sigma(k)} \Delta \vec{u}_{0|k} + G_{\sigma(k)}^p \Delta \vec{u}_{k|k-1}^p$$

where

$$G_{\sigma(k)}^p = \begin{bmatrix} g_2^{\sigma(k)} & g_3^{\sigma(k)} & \dots & g_m^{\sigma(k)} \end{bmatrix}.$$

The cost function of (2.7) can then be re-written as following

$$\begin{aligned}
J_k &= F(x_{k+N|k}) + \sum_{i=0}^{N-1} (\|x_{k+i+1|k}\|_q^2 + \|\Delta \tilde{u}_{k+i|k}\|_r^2) \\
&= x_{k+N|k}^T \bar{P}_{\sigma(k+N)} x_{k+N|k} + \sum_{i=0}^{N-1} (\|x_{k+i+1|k}\|_q^2 + \|\Delta \tilde{u}_{k+i|k}\|_r^2) \\
&= \|\vec{X}_{k+1|k}\|_{Q_{\sigma(k)}}^2 + \|\Delta \vec{u}_{0|k}\|_{R_u}^2 + \sum_{i=1}^{m-1} \|\Delta \vec{u}_{k,i|k}\|_{R_u}^2 \\
&= \|\Phi x_{k|k} + g_1^{\sigma(k)} \Delta \vec{u}_{0|k} + G_{\sigma(k)}^p \Delta \vec{u}_{k|k-1}^p\|_{Q_{\sigma(k)}}^2 + \|\Delta \vec{u}_{0|k}\|_{R_u}^2 + \sum_{i=1}^{m-1} \|\Delta \vec{u}_{k,i|k}\|_{R_u}^2 \\
&= \Delta \vec{u}_{0|k}^T \Theta_u^{\sigma(k)} \Delta \vec{u}_{0|k} + 2 \Delta \vec{u}_{0|k}^T \Theta_x^{\sigma(k)} \begin{bmatrix} x_{k|k} & \Delta \vec{u}_{k|k-1}^p \end{bmatrix}^T + \Omega_k
\end{aligned}$$

where

$$Q_{\sigma(k)} = \begin{bmatrix} q & 0 & \cdots & 0 \\ 0 & q & \ddots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & q + \bar{P}_{\sigma(k+N)} \end{bmatrix}, \quad R_u = \begin{bmatrix} r & 0 & \cdots & 0 \\ 0 & r & \ddots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & r \end{bmatrix}$$

$$\Theta_u^{\sigma(k)} = (g_1^{\sigma(k)})^T Q_{\sigma(k)} g_1^{\sigma(k)} + R_u, \quad \Theta_x^{\sigma(k)} = \begin{bmatrix} (g_1^{\sigma(k)})^T Q_{\sigma(k)} \Phi & (g_1^{\sigma(k)})^T Q_{\sigma(k)} G_{\sigma(k)}^p \end{bmatrix}$$

Ω_k contains the remaining terms which do not depend on $\Delta \vec{u}_{0|k}$. Therefore, if no constraints are active, minimize J_k with respect to $\Delta \vec{u}_{0|k}$, we can obtain the MMPC control law as a linear periodic state feedback of

$$\Delta \vec{u}_{0|k} = \tilde{K}_{\sigma(k)} \xi_k = \tilde{K}_{\sigma(k)} \begin{bmatrix} x_k \\ \Delta \vec{u}_{k|k-1}^p \end{bmatrix}, \quad (3.7)$$

where

$$\tilde{K}_{\sigma(k)} = -[(g_1^{\sigma(k)})^T Q_{\sigma(k)} g_1^{\sigma(k)} + R_u]^{-1} \begin{bmatrix} (g_1^{\sigma(k)})^T Q_{\sigma(k)} \Phi & (g_1^{\sigma(k)})^T Q_{\sigma(k)} G_{\sigma(k)}^p \end{bmatrix}$$

Here we have introduced the augmented state vector

$$\xi_k = \begin{bmatrix} x_k \\ \Delta \vec{u}_{k|k-1}^p \end{bmatrix}. \quad (3.8)$$

Using the augmented state vector introduced in (3.8), the dynamics of the plant operating under MMPC can be expressed as

$$\xi_{k+1} = \tilde{A}\xi_k + \tilde{B}_{\sigma(k)}\Delta \vec{u}_{0|k} \quad (3.9)$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A_u \end{bmatrix} \quad \tilde{B}_{\sigma(k)} = \begin{bmatrix} B_{\sigma(k)} & 0 \\ 0 & B_u \end{bmatrix}$$

$$A_u = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & & 0 \end{bmatrix} \quad B_u = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ I \end{bmatrix}$$

3.2 Closed-loop Monodromy Matrix of MMPC

An immediate application of the augmented state space model for MMPC is to compute the closed-loop monodromy matrix of MMPC.

Theorem 3.2.1 *The closed-loop monodromy matrix of the system (3.9), when any MMPC control law of the form (3.7) is applied, is given by*

$$\Psi_{\sigma(k)} = \tilde{\Phi}_{\sigma(k+m-1)} \tilde{\Phi}_{\sigma(k+m-2)} \cdots \tilde{\Phi}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \quad (3.10)$$

where

$$\tilde{\Phi}_{\sigma(k+i)} = \tilde{A} + \tilde{B}_{\sigma(k+i)}\tilde{K}_{\sigma(k+i)}, \quad i = 0, 1, \dots, m-1$$

Proof:

(3.9) together with (3.7), enable one to re-cast the (unconstrained) MMPC formulation in the standard periodic control framework. Now we can get the closed-loop state transition formula in the augmented state space:

$$\xi_{k+1} = (\tilde{A} + \tilde{B}_{\sigma(k)}\tilde{K}_{\sigma(k)})\xi_k = \tilde{\Phi}_{\sigma(k)}\xi_k$$

Thus the remaining closed-loop state transitions within one updating cycle T are

$$\begin{aligned} \xi_{k+2} &= (\tilde{A} + \tilde{B}_{\sigma(k+1)}\tilde{K}_{\sigma(k+1)})\xi_{k+1} = \tilde{\Phi}_{\sigma(k+1)}\tilde{\Phi}_{\sigma(k)}\xi_k \\ \vdots &\quad \quad \quad \vdots \\ \xi_{k+m} &= (\tilde{A} + \tilde{B}_{\sigma(k+m-1)}\tilde{K}_{\sigma(k+m-1)})\xi_{k+m-1} = \tilde{\Phi}_{\sigma(k+m-1)}\tilde{\Phi}_{\sigma(k+m-2)} \cdots \tilde{\Phi}_{\sigma(k+1)}\tilde{\Phi}_{\sigma(k)}\xi_k \end{aligned}$$

So the closed-loop monodromy matrix is given by

$$\Psi_{\sigma(k)} = \tilde{\Phi}_{\sigma(k+m-1)}\tilde{\Phi}_{\sigma(k+m-2)} \cdots \tilde{\Phi}_{\sigma(k+1)}\tilde{\Phi}_{\sigma(k)}$$

■

If all the eigenvalues of $\Psi_{\sigma(k)}$ are within the unit cycle, the closed-loop under MMPC will be stable.

Note that the dimensions of augmented state vector ξ_k , matrices $\tilde{A}$, $\tilde{B}_{\sigma(k)}$ and the controller gain $\tilde{K}_{\sigma(k)}$ all depend on the parameter N_u . As an illustration, consider

the numerical example (2.17). When $N_u = 1$, $\Delta \vec{u}_{k|k-1}^p$ vanishes, therefore,

$$\tilde{A} = \begin{bmatrix} 0.9311 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9394 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8465 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8825 & 0 & 0 \\ 0.2660 & 0 & 0.5643 & 0 & 1.0000 & 0 \\ 0 & 0.4697 & 0 & 0.4412 & 0 & 1.0000 \end{bmatrix},$$

$$\tilde{B}_1 = \begin{bmatrix} 0.2413 \\ 0.2423 \\ 0 \\ 0 \\ 0.0689 \\ 0.1212 \end{bmatrix}, \quad \tilde{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 0.2303 \\ 0.2350 \\ 0.1535 \\ 0.1175 \end{bmatrix}$$

and the periodic controller gains are

$$\tilde{K}_1 = \begin{bmatrix} 0.7289 & -2.4886 & 0.9829 & -1.9283 & 0.2203 & -1.0224 \end{bmatrix}$$

$$\tilde{K}_2 = \begin{bmatrix} -1.1941 & 0.2884 & -1.9604 & 0.0983 & -0.9706 & -0.2091 \end{bmatrix}$$

It's nominal closed loop eigenvalues are

$$\begin{bmatrix} 0.8423 & 0.8423 & 0.7418 & 0.7418 & 0.5429 & 0.5429 \end{bmatrix},$$

which are all within the unit cycle.

With $N_u = 2$, the matrices becomes

$$\tilde{A} = \begin{bmatrix} 0.9311 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9394 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8465 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8825 & 0 & 0 & 0 \\ 0.2660 & 0 & 0.5643 & 0 & 1.0000 & 0 & 0 \\ 0 & 0.4697 & 0 & 0.4412 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{B}_1 = \begin{bmatrix} 0.2413 & 0 \\ 0.2423 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.0689 & 0 \\ 0.1212 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tilde{B}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.2303 & 0 \\ 0.2350 & 0 \\ 0.1535 & 0 \\ 0.1175 & 0 \\ 0 & 1 \end{bmatrix}$$

and the periodic gains are

$$\tilde{K}_1 = \begin{bmatrix} 0.2321 & -2.1944 & 0.2011 & -1.7437 & -0.1031 & -0.9887 & -0.3112 \\ 0.6025 & -1.7807 & 0.8104 & -1.2808 & 0.2835 & -0.4156 & -0.1221 \end{bmatrix}$$

$$\tilde{K}_2 = \begin{bmatrix} -0.9060 & -0.4657 & -1.5533 & -0.4679 & -0.8347 & -0.4572 & -0.2593 \\ -0.9148 & 0.3822 & -1.3549 & 0.2225 & -0.4004 & 0.1032 & -0.1333 \end{bmatrix}$$

The closed-loop eigenvalues are

$$\begin{bmatrix} 0.8423 & 0.8423 & 0.7418 & 0.7418 & 0.5429 & 0.5429 & 0.0807 \end{bmatrix}$$

There are 7 eigenvalues when $N_u = 2$, while only 6 when $N_u = 1$. With $N_u = 2$, the length of $\Delta \bar{u}_{k|k-1}^p$ in the augmented state ξ_k is 1, while the length is 0 when $N_u = 1$.

Table 3.1: Closed-loop poles for MMPC controlled plant with increasing N_u

N_u	$Pole1$	$Pole2$	$Pole3$	$Pole4$	$Pole5$	$Pole6$	$Pole7$	$Pole8$	$Pole9$	$Pole10$
1	0.8423	0.8423	0.7418	0.7418	0.5429	0.5429				
2	0.8423	0.8423	0.7418	0.7418	0.5429	0.5429	0.0807			
3	0.8423	0.8423	0.7418	0.7418	0.5429	0.5429	0.2363	0.0017		
4	0.8423	0.8423	0.7418	0.7418	0.5429	0.5429	0.3803	0.0145	0.0001	
5	0.8423	0.8423	0.7418	0.7418	0.5429	0.5429	0.4904	0.0494	0.0012	0.0000

For comparison, Table 3.1 list all the poles of closed-loop system for N_u from 1 to 5. Since all the eigenvalues are within the unit circle, the closed-loop system is stable.

3.3 A Formula for the MMPC Cost

The quadratic cost

$$J_k = \sum_{i=0}^{\infty} (\|\xi_{k+i+1}\|_{\bar{q}}^2 + \|\Delta \vec{u}_{0|k+i}\|_{\bar{r}}^2) \quad (3.11)$$

when control law of the form (3.7) is applied, is given by the following proposition:

Theorem 3.3.1 *The value of the cost (3.11) for the system (3.9), when any stabilising linear periodic state feedback of the form (3.7) is applied, is given by*

$$J_k = J_{\xi,k} + J_{u,k} = \xi_k^T P_{\xi+u,\sigma(k)} \xi_k \quad (3.12)$$

where $J_{\xi,k} = \xi_k^T P_{\xi,\sigma(k)} \xi_k$ and $J_{u,k} = \xi_k^T P_{u,\sigma(k)} \xi_k$, and $P_{\xi,\sigma(k)}$, $P_{u,\sigma(k)}$ and $P_{\xi+u,\sigma(k)}$

are, respectively, solutions of the following Lyapunov equations:

$$P_{\xi, \sigma(k)} = \Psi_{\sigma(k)}^T P_{\xi, \sigma(k)} \Psi_{\sigma(k)} + \bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)} \quad (3.13)$$

$$P_{u, \sigma(k)} = \Psi_{\sigma(k)}^T P_{u, \sigma(k)} \Psi_{\sigma(k)} + \bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)} \quad (3.14)$$

$$\begin{aligned} P_{\xi+u, \sigma(k)} &= \Psi_{\sigma(k)}^T P_{\xi+u, \sigma(k)} \Psi_{\sigma(k)} + \bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)} \\ &\quad + \bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)} \end{aligned} \quad (3.15)$$

where

$$\begin{aligned} \Psi_{\sigma(k)} &= \tilde{\Phi}_{\sigma(k+m-1)} \cdots \tilde{\Phi}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \\ \tilde{\Phi}_{\sigma(k)} &= \tilde{A} + \tilde{B}_{\sigma(k)} \tilde{K}_{\sigma(k)} \\ \bar{\Phi}_{\sigma(k)} &= \begin{bmatrix} \tilde{\Phi}_{\sigma(k)} \\ \tilde{\Phi}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \\ \vdots \\ \tilde{\Phi}_{\sigma(k+m-1)} \cdots \tilde{\Phi}_{\sigma(k)} \end{bmatrix} \\ \bar{K}_{\sigma(k)} &= \begin{bmatrix} \tilde{K}_{\sigma(k)} \\ \tilde{K}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \\ \vdots \\ \tilde{K}_{\sigma(k+m-1)} \tilde{\Phi}_{\sigma(k+m-2)} \cdots \tilde{\Phi}_{\sigma(k)} \end{bmatrix} \\ Q &= \text{diag}(\tilde{q}, \tilde{q}, \dots, \tilde{q}) \quad \text{and} \\ R &= \text{diag}(\tilde{r}, \tilde{r}, \dots, \tilde{r}) \end{aligned}$$

Proof:

The closed-loop dynamics of the system (3.9), when stabilising linear periodic state feedback of the form (3.7) is applied, is given by

$$\xi_{k+1} = \tilde{A} \xi_k + \tilde{B}_{\sigma(k)} \Delta \vec{u}_{0|k} = (\tilde{A} + \tilde{B}_{\sigma(k)} \tilde{K}_{\sigma(k)}) \xi_k$$

3.3 A Formula for the MMPC Cost

42

or in general

$$\xi_{k+i+1} = \tilde{\Phi}_{\sigma(k+i)} \xi_{k+i}, \quad i = 0, 1, \dots$$

where

$$\tilde{\Phi}_{\sigma(k+i)} = \tilde{A} + \tilde{B}_{\sigma(k+i)} \tilde{K}_{\sigma(k+i)}$$

Then

$$\begin{bmatrix} \xi_{k+jm+1} \\ \xi_{k+jm+2} \\ \vdots \\ \xi_{k+jm+m} \end{bmatrix} = \begin{bmatrix} \tilde{\Phi}_{\sigma(k)} \\ \tilde{\Phi}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \\ \vdots \\ \underbrace{\tilde{\Phi}_{\sigma(k+m-1)} \cdots \tilde{\Phi}_{\sigma(k)}}_{\tilde{\Phi}_{\sigma(k)}} \end{bmatrix} \xi_{k+jm}$$

and

$$\begin{aligned} \begin{bmatrix} \Delta \vec{u}_{0|k+jm} \\ \Delta \vec{u}_{0|k+jm+1} \\ \vdots \\ \Delta \vec{u}_{0|k+jm+m-1} \end{bmatrix} &= \begin{bmatrix} \tilde{K}_{\sigma(k)} \xi_{k+jm} \\ \tilde{K}_{\sigma(k+1)} \xi_{k+jm+1} \\ \vdots \\ \tilde{K}_{\sigma(k+m-1)} \xi_{k+jm+m-1} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{K}_{\sigma(k)} \\ \tilde{K}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)} \\ \vdots \\ \underbrace{\tilde{K}_{\sigma(k+m-1)} \tilde{\Phi}_{\sigma(k+m-2)} \cdots \tilde{\Phi}_{\sigma(k)}}_{\tilde{K}_{\sigma(k)}} \end{bmatrix} \xi_{k+jm} \end{aligned}$$

Thus,

$$\begin{aligned}
J_{\xi,k} &= \sum_{i=0}^{\infty} \|\xi_{k+i+1}\|_{\tilde{q}}^2 \\
&= \sum_{j=0}^{\infty} \sum_{i=1}^m \|\xi_{k+jm+i}\|_{\tilde{q}}^2 \\
&= \sum_{j=0}^{\infty} \xi_{k+jm}^T \bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)} \xi_{k+jm} \\
&= \xi_k^T \left[\sum_{j=0}^{\infty} (\Psi_{\sigma(k)}^j)^T [\bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)}] \Psi_{\sigma(k)}^j \right] \xi_k \\
&= \xi_k^T P_{\xi,\sigma(k)} \xi_k
\end{aligned}$$

where

$$\begin{aligned}
Q &= \text{diag}(\tilde{q}, \tilde{q}, \dots, \tilde{q}) \\
\Psi_{\sigma(k)} &= \tilde{\Phi}_{\sigma(k+m-1)} \cdots \tilde{\Phi}_{\sigma(k+1)} \tilde{\Phi}_{\sigma(k)}
\end{aligned}$$

and

$$P_{\xi,\sigma(k)} = \sum_{j=0}^{\infty} (\Psi_{\sigma(k)}^j)^T [\tilde{\Phi}_{\sigma(k)}^T Q \tilde{\Phi}_{\sigma(k)}] \Psi_{\sigma(k)}^j$$

is a convergent series since the controller $\tilde{K}_{\sigma(k)}$ is stabilizing. Thus $P_{\xi,\sigma(k)}$ can be computed by solving the following Lyapunov equation

$$P_{\xi,\sigma(k)} = \Psi_{\sigma(k)}^T P_{\xi,\sigma(k)} \Psi_{\sigma(k)} + \bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)}$$

Similarly, the sum of the control increments can be computed as

$$\begin{aligned}
J_{u,k} &= \sum_{i=0}^{\infty} \|\Delta \vec{u}_{0|k+i}\|_{\tilde{r}}^2 \\
&= \sum_{j=0}^{\infty} \sum_{i=0}^{m-1} \|\Delta \vec{u}_{0|k+jm+i}\|_{\tilde{r}}^2 \\
&= \sum_{j=0}^{\infty} \sum_{i=0}^{m-1} \|\bar{K}_{\sigma(k+i)} \xi_{k+jm+i}\|_{\tilde{r}}^2 \\
&= \sum_{j=0}^{\infty} \xi_{k+jm}^T \bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)} \xi_{k+jm} \\
&= \xi_k^T \sum_{j=0}^{\infty} (\Psi_{\sigma(k)}^j)^T [\bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)}] \Psi_{\sigma(k)}^j \xi_k \\
&= x_k^T P_{u,\sigma(k)} x_k
\end{aligned}$$

where

$$R = \text{diag}(\tilde{r}, \tilde{r}, \dots, \tilde{r})$$

and

$$P_{u,\sigma(k)} = \sum_{i=0}^{\infty} (\Psi_{\sigma(k)}^i)^T [\bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)}] \Psi_{\sigma(k)}^i$$

which can be computed by solving the Lyapunov equation

$$P_{u,\sigma(k)} = \Psi_{\sigma(k)}^T P_{u,\sigma(k)} \Psi_{\sigma(k)} + \bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)}$$

Finally, let

$$\begin{aligned} P_{\xi+u,\sigma(k)} &= P_{\xi,\sigma(k)} + P_{u,\sigma(k)} \\ &= \sum_{j=0}^{\infty} (\Psi_{\sigma(k)}^j)^T [\bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)}] \Psi_{\sigma(k)}^j + \sum_{i=0}^{\infty} (\Psi_{\sigma(k)}^i)^T [\bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)}] \Psi_{\sigma(k)}^i \end{aligned}$$

and it is clear that $P_{\xi+u,\sigma(k)}$ can be computed as the solution of the Lyapunov equation

$$P_{\xi+u,\sigma(k)} = \Psi_{\sigma(k)}^T P_{\xi+u,\sigma(k)} \Psi_{\sigma(k)} + \bar{\Phi}_{\sigma(k)}^T Q \bar{\Phi}_{\sigma(k)} + \bar{K}_{\sigma(k)}^T R \bar{K}_{\sigma(k)}$$

■

The quadratic cost

$$J_k = \sum_{i=0}^{\infty} (\|x_{k+i+1}\|_q^2 + \|\Delta \tilde{u}_{k+i}\|_r^2) \quad (3.16)$$

when MMPC control law is applied, is given by (3.11) if

$$\tilde{q} = \begin{bmatrix} q & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{r} = \begin{bmatrix} r & 0 \\ 0 & 0 \end{bmatrix} \quad (3.17)$$

and the corresponding initial conditions hold on ξ_k . For example, at the beginning of an MMPC run it may be appropriate to set

$$\xi_k = \begin{bmatrix} x_k^T & (\Delta \tilde{u}_{k|k-1}^p)^T \end{bmatrix}^T = \begin{bmatrix} x_k^T & 0 \end{bmatrix}^T$$

Remark 3.1 *If the second part of ξ_k , namely $\Delta \tilde{u}_{k|k-1}^p$, were included in the optimization, so that the optimal cost became a function of x_k only, then the optimal cost, and the optimal solution, would be the same as that obtained with ‘scheme 1’ in our earlier papers [37], namely it would correspond to the cost resulting from*

allowing each ‘agent’ to optimize all future inputs rather than just ‘its own’ input.

3.4 An Illustrative Example

In this section, numerical examples will be given to illustrate how the cost formula for MMPC can be used for evaluating the effect of design choice available in MMPC, such as the control horizon of N_u and updating sequence, on the closed-loop performance.

The cost formula for MMPC is calculated as (3.11) with (3.17) and initial condition of $\xi_k = \begin{bmatrix} x_k^T & 0 \end{bmatrix}^T$. Hence, only the upper-left $n \times n$ (n is the dimension of x_k) sub-matrices are relevant in the cost computation. To be specific, the sub-matrices are $\hat{P}_{\sigma(k)}$, $\hat{P}_{x,\sigma(k)}$ and $\hat{P}_{u,\sigma(k)}$ as shown below

$$P_{\xi+u,\sigma(k)} = \begin{bmatrix} \hat{P}_{\sigma(k)} & \star \\ \star & \star \end{bmatrix}, \quad P_{\xi,\sigma(k)} = \begin{bmatrix} \hat{P}_{x,\sigma(k)} & \star \\ \star & \star \end{bmatrix}, \quad P_{u,\sigma(k)} = \begin{bmatrix} \hat{P}_{u,\sigma(k)} & \star \\ \star & \star \end{bmatrix}$$

Therefore, the quadratic cost of MMPC can be computed as

$$J_{\sigma(k)} = J_{x,\sigma(k)} + J_{u,\sigma(k)} = x_k^T \hat{P}_{\sigma(k)} x_k$$

where $J_{x,\sigma(k)} = x_k^T \hat{P}_{x,\sigma(k)} x_k$ and $J_{u,\sigma(k)} = x_k^T \hat{P}_{u,\sigma(k)} x_k$.

Now we have a way to compare different MMPC schemes. In other words, given two MMPC schemes, whose costs are $J_i = x_k^T \hat{P}_i x_k$ and $J_j = x_k^T \hat{P}_j x_k$, then

$$J_i - J_j = x_k^T (\hat{P}_i - \hat{P}_j) x_k$$

So, from the eigenvalues of $(\hat{P}_i - \hat{P}_j)$, we could compare the two MMPC schemes⁴.

⁴For the case when $(\hat{P}_i - \hat{P}_j)$ is positive or negative definite, one can also use $\text{trace}(\hat{P}_i) - \text{trace}(\hat{P}_j)$ to compare the two MMPC schemes

Consider the following two-input-two-output continuous-time plant

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{7s+1} & \frac{1}{3s+1} \\ \frac{2}{8s+1} & \frac{1}{4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

we chose the sampling time to be $T = 1s$. For MMPC schemes, the states were measured at $T/m = 0.5s$ with u_1 and u_2 alternatively applied at $0.5s$ intervals, but each held constant over a period of $T = 1s$, that is, u_1 is updated at times $(0, 1s, 2s, \dots)$ and u_2 is updated at times $(0.5s, 1.5s, \dots)$ (for the updating sequence of $u_1, u_2, u_1, u_2, \dots$). For all the results listed below, the tuning parameters for MMPC are: $q = C^T C$ and $r = 1$.

- **The effect of tuning parameter N_u in MMPC**

For MMPC, N_u is an important tuning parameters. With the MMPC cost formula, we could compute the cost and predict the performance difference with different N_u . Table 3.2 shows the eigenvalues of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ for various combinations of i and j ($i, j \in \{1, 2, 3, 4, 5\}$) when the updating sequence is $\{u_2, u_1\}$, while Table 3.3 shows the same for the updating sequence $\{u_1, u_2\}$. It could be seen that all the matrices $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=1)})$, $i = 2, 3, 4, 5$, can be considered as practically positive definite. This does not mean that $N_u = 1$ is the best choice for MMPC. Usually, we need to make the prediction horizon of $N = m(N_u - 1) + 1$ in MMPC be sufficiently long to ensure the feasibility of the constrained optimization problem. From Table 3.2 and Table 3.3, we also could find that matrices $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ ($i > j$) are positive dominant, which indicates that in most cases, as N_u increases, the cost gets bigger. To illustrate, we generate a simulation scenario by adding a step input disturbance to the plant and see how their performance cost will be with different N_u based on this initial condition. For the reason of simplicity, we just use the case of $(\hat{P}_{(N_u=3)} - \hat{P}_{(N_u=2)})$ with updating sequence $\{u_2, u_1\}$ to present a representative cost comparison.

$$\hat{P}_{(N_u=2)} = \begin{bmatrix} 10.7921 & -15.8574 & 15.2578 & -11.2006 & 5.5397 & -4.5091 \\ -15.8574 & 29.9903 & -22.4028 & 22.0589 & -7.6298 & 10.0788 \\ 15.2578 & -22.4028 & 23.4664 & -16.7490 & 9.3788 & -7.2344 \\ -11.2006 & 22.0589 & -16.7490 & 16.9334 & -5.9825 & 8.2598 \\ 5.5397 & -7.6298 & 9.3788 & -5.9825 & 6.0921 & -2.8997 \\ -4.5091 & 10.0788 & -7.2344 & 8.2598 & -2.8997 & 6.5178 \end{bmatrix}$$

$$\hat{P}_{(N_u=3)} = \begin{bmatrix} 11.3758 & -17.2878 & 16.0794 & -12.2463 & 5.8222 & -4.9191 \\ -17.2878 & 33.4914 & -24.4157 & 24.6173 & -8.3224 & 11.0798 \\ 16.0794 & -24.4157 & 24.6227 & -18.2206 & 9.7764 & -7.8113 \\ -12.2463 & 24.6173 & -18.2206 & 18.8026 & -6.4889 & 8.9905 \\ 5.8222 & -8.3224 & 9.7764 & -6.4889 & 6.2288 & -3.0984 \\ -4.9191 & 11.0798 & -7.8113 & 8.9905 & -3.0984 & 6.8024 \end{bmatrix}$$

and the matrix $(\hat{P}_{(N_u=3)} - \hat{P}_{(N_u=2)})$ has eigenvalues

$$\begin{bmatrix} -0.0040 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 7.5355 \end{bmatrix}$$

which means that in most cases the quadratic cost of $N_u = 3$ will be bigger than that of $N_u = 2$.

Table 3.4 and Table 3.5 list all the cost details for different N_u with updating sequence of $\{u_2, u_1\}$ and $\{u_1, u_2\}$ respectively. It could be observed that the overall cost has increased with increasing N_u , which could be predicted from the eigenvalues of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$, as it shows positive dominant values if $i > j$. More detailed investigation shows that the tracking error ($J_{x,mmpc}$) decreases as N_u increased from 1 to 2 before increasing again when

$N_u = 3$, while the control activities ($J_{u,mmpc}$) increases with increasing N_u . The ratio between tracking error cost of ($J_{x,mmpc}$) and the overall cost of (J_{mmpc}) decreases from 1 to 3 before rising again from $N_u = 4$. Figure 3.1 and Figure 3.2 show their corresponding closed-loop performance.

Based on the detailed cost comparison, we could have some idea about choosing proper N_u under certain situation so that the overall performance is the best acceptable.

- **The effect of different updating sequence**

As it is stated before, MMPC is a periodic control scheme with flexible choice of updating patterns, a typical one of them is the updating sequence. This section will show how the updating sequence effects the cost and the closed-loop performance with the same system setting. We will still use the step input disturbance as a specific simulation scenario.

Table 3.6 shows the eigenvalues of $(\hat{P}_1 - \hat{P}_2)$ as N_u varies, where $\hat{P}_1$ represents the cost weighting for updating sequence of $\{u_1, u_2\}$ while $\hat{P}_2$ represents the cost weighting for updating sequence of $\{u_2, u_1\}$. From the table, it could be observed that the eigenvalues of $(\hat{P}_1 - \hat{P}_2)$ is indefinite, which means that one updating sequence is not definitely better than the other, it depends on the specific application scenarios. Figure 3.3 compares the closed-loop performance with the two updating sequence and $N_u = 5$, it could be seen that with updating sequence of $\{u_1, u_2\}$, the overshoot of y_1 is greater than that of updating sequence of $\{u_2, u_1\}$, while the former has less undershoot. For output channel y_2 , the updating sequence of $\{u_1, u_2\}$ shows bigger undershoot and less overshoot than the updating sequence of $\{u_2, u_1\}$. It is hard to decide which one is apparently better simply from the simulation plots. For completeness, the numerical value for $\hat{P}_1$, $\hat{P}_2$ and the matrix decomposition

of $(\hat{P}_1 - \hat{P}_2)$ with $N_u = 5$ are also listed below:

$$\hat{P}_1 = \begin{bmatrix} 17.8694 & -19.2913 & 25.8928 & -13.1706 & 9.7211 & -4.5535 \\ -19.2913 & 27.8821 & -27.6436 & 19.9475 & -9.8537 & 8.2874 \\ 25.8928 & -27.6436 & 39.4833 & -19.7753 & 15.7237 & -7.3165 \\ -13.1706 & 19.9475 & -19.7753 & 14.9889 & -7.3057 & 6.7981 \\ 9.7211 & -9.8537 & 15.7237 & -7.3057 & 8.6671 & -2.9791 \\ -4.5535 & 8.2874 & -7.3165 & 6.7981 & -2.9791 & 5.6610 \end{bmatrix}$$

$$\hat{P}_2 = \begin{bmatrix} 13.4666 & -22.0546 & 18.9613 & -15.5909 & 6.7899 & -6.0966 \\ -22.0546 & 44.3442 & -30.9841 & 32.2257 & -10.5273 & 13.7515 \\ 18.9613 & -30.9841 & 28.5948 & -22.8284 & 11.1101 & -9.4325 \\ -15.5909 & 32.2257 & -22.8284 & 24.1340 & -8.0354 & 10.8598 \\ 6.7899 & -10.5273 & 11.1101 & -8.0354 & 6.6766 & -3.6423 \\ -6.0966 & 13.7515 & -9.4325 & 10.8598 & -3.6423 & 7.4547 \end{bmatrix}$$

and the eigenvalues of $(\hat{P}_1 - \hat{P}_2)$ is

$$\begin{bmatrix} -28.3386 & -0.0710 & -0.0001 & 0.0000 & 0.0905 & 18.2002 \end{bmatrix}$$

The cost difference between $\hat{P}_1$ and $\hat{P}_2$ in this case is 0.3599, which means that the updating sequence of $\{u_2, u_1\}$ is slightly better than that of $\{u_1, u_2\}$.

3.5 Conclusion

In this chapter, we made further analysis of the MMPC scheme. It was shown that the (unconstrained) MMPC can be cast in the standard periodic control framework and this has enabled a systematic analysis of its closed-loop properties. We also derived formula for computing the MMPC cost and use it as a tool to evaluate MMPC schemes with varying degrees of freedom represented by the tuning param-

eter N_u . The cost formula is also useful in evaluating the performance of MMPC with different updating patterns. An illustrative example has been employed to show the unconstrained MMPC as a linear state feedback and its closed-loop properties with different tuning parameter N_u . Furthermore, the illustrative example has also been employed to demonstrate the usage of cost formula in selecting tuning parameters and updating patterns.

The analysis carried out in Section 3.3 suggests yet another method to compute the terminal cost $F(x_{k+N|k})$ to ensure nominal stability of MMPC in addition to that presented in Proposition 1. A family of MMPC design may be obtained by optimizing the cost function (3.11) subject to the model (3.9) by choosing appropriate $\tilde{q}$ and $\tilde{r}$ matrices satisfying conditions for the existence of a stabilizing solution of resulting Discrete-time Periodic Riccati Equation (DPRE).

Table 3.2: MMPC: Eigenvalue of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ with input disturbance at 5.0s

i	j	Eigenvalues of $\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)}$					
2	1	0.0000	0.0000	0.0000	0.0000	0.0000	2.7503
3	1	0.0000	0.0000	0.0000	0.0000	0.0002	10.2815
4	1	0.0000	0.0000	0.0000	0.0000	0.0022	21.0558
5	1	0.0000	0.0000	0.0000	0.0000	0.0089	33.6200
3	2	-0.0040	0.0000	0.0000	0.0000	0.0000	7.5355
4	2	-0.0077	0.0000	0.0000	0.0000	0.0000	18.3154
5	2	-0.0070	0.0000	0.0000	0.0000	0.0001	30.8855
4	3	-0.0083	0.0000	0.0000	0.0000	0.0000	10.7846
5	3	-0.0144	0.0000	0.0000	0.0000	0.0000	23.3615
5	4	-0.0093	0.0000	0.0000	0.0000	0.0000	12.5802

Figure 3.1: Step input disturbance rejection with increasing N_u , $\{u_2, u_1\}$ Figure 3.2: Step input disturbance rejection with increasing N_u , $\{u_1, u_2\}$

Table 3.3: MMPC: Eigenvalue of $(\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)})$ with input disturbance at 5.5s

i	j	Eigenvalues of $\hat{P}_{(N_u=i)} - \hat{P}_{(N_u=j)}$					
2	1	0.0000	0.0000	0.0000	0.0000	0.0000	2.1189
3	1	0.0000	0.0000	0.0000	0.0000	0.0003	8.5939
4	1	0.0000	0.0000	0.0000	0.0000	0.0054	17.7589
5	1	0.0000	0.0000	0.0000	0.0000	0.0270	28.0080
3	2	-0.0126	0.0000	0.0000	0.0000	0.0000	6.4879
4	2	-0.0262	0.0000	0.0000	0.0000	0.0000	15.6716
5	2	-0.0241	0.0000	0.0000	0.0000	0.0003	25.9399
4	3	-0.0320	0.0000	0.0000	0.0000	0.0000	9.2020
5	3	-0.0566	0.0000	0.0000	0.0000	0.0000	19.4974
5	4	-0.0376	0.0000	0.0000	0.0000	0.0000	10.3083

Table 3.4: Cost Comparison of MMPC with increasing N_u , updating sequence $\{u_2, u_1\}$

N_u	1	2	3	4	5
J_{mmpc}	2.5152	2.5897	2.7392	2.9063	3.0693
$J_{x,mmpc}$	0.9417	0.8272	0.8453	0.9298	1.0450
$J_{u,mmpc}$	1.5736	1.7624	1.8939	1.9766	2.0243
$J_{x,mmpc}/J_{mmpc}$	37.44%	31.94%	30.86%	31.99%	34.05%
$J_{x,mmpc}/J_{u,mmpc}$	0.5985	0.4693	0.4463	0.4704	0.5162

Table 3.5: Cost Comparison of MMPC with increasing N_u , updating sequence $\{u_1, u_2\}$

N_u	1	2	3	4	5
J_{mmpc}	2.6629	2.7841	3.0262	3.2523	3.4292
$J_{x,mmpc}$	1.0781	0.9516	0.9564	1.0262	1.1214
$J_{u,mmpc}$	1.5847	1.8325	2.0699	2.2260	2.3078
$J_{x,mmpc}/J_{mmpc}$	40.49%	34.18%	31.60%	31.55%	32.70%
$J_{x,mmpc}/J_{u,mmpc}$	0.6803	0.5193	0.4621	0.4610	0.4859

Table 3.6: MMPC: Eigenvalue of $(\hat{P}_1 - \hat{P}_2)$ with different N_u

N_u	Eigenvalues of $(\hat{P}_1 - \hat{P}_2)$					
1	-9.4274	-0.0069	0.0000	0.0000	0.0042	4.9050
2	-11.3972	-0.0008	-0.0002	0.0001	0.0014	6.2401
3	-15.9939	-0.0190	-0.0001	0.0001	0.0289	9.7714
4	-21.9017	-0.0438	-0.0002	0.0001	0.0603	14.0665
5	-28.3386	-0.0710	-0.0001	0.0000	0.0905	18.2002

Figure 3.3: Effects of updating sequence, step input disturbance at 5.5s, $N_u = 5$, solid: $\{u_1, u_2\}$, dashed: $\{u_2, u_1\}$.

Chapter 4

Multiplexed MPC for Wafer Baking Process

The main aim of this chapter is to apply Multiplexed MPC to a real application and demonstrate that MMPC offers real advantage over traditional MPC (SMPC) in rejecting disturbances more effectively due to its control strategy.

In this chapter, the MMPC scheme is applied to a wafer baking process normally encountered in lithography sequence (see Figure 4.1) of semiconductor manufacturing. The lithography sequence is briefly described first followed by the physical model development of a multizone bake plate. In Section 4.3, open-loop tests were conducted to collect the input/output data for parameter estimation of a three-zone bake plate. Section 4.4 presents the experimental results, which compare the closed-loop performance of MMPC and SMPC when a load disturbance occurs. The experimental results showed that MMPC indeed offered advantage over traditional MPC in that it reject the disturbance faster. Experimental results where constraints became active are also shown, demonstrating that MMPC succeeded in maintaining stability in the presence of constraints and disturbances. Section 4.5 compared the computational scalability of MMPC and SMPC and Section 4.6 showed the effects of sampling rate on disturbance rejection performance. Finally,

conclusion is given in Section 4.7.

Figure 4.1: The lithography sequence, extracted from [1]

4.1 The Lithography Sequence

Lithography is a microfabrication technique used to make integrated circuits and is regarded as the most important process in semiconductor manufacturing due to its strong influence on cost and performance of a microchip. Typical lithography sequence is illustrated in Figure 4.1. It usually begins with a priming step to promote adhesion of the polymer photoresist material to the substrate. A thin layer of photoresist is spin-coated on the wafer surface. The solvent is evaporated from the photoresist by a soft baking process. After patterning with deep UV radiation, a post-exposure bake (PEB) process is used to promote a reaction that alters the solubility of the photoresist in the exposed areas. A subsequent chemical develop step then removes the photoresist materials while keeping the non-exposed

areas in place. The developed resist is then baked to promote etching stability.

In the lithography sequence, the most important variable to be controlled is the minimum feature size dimension or critical dimension (CD) since it has the most impact on device speed and performance. This variable is affected by various processes especially baking processes, which is also refereed as thermal processing in the semiconductor manufacturing [52, 53, 54, 55]. Thermal processing of semiconductor wafers is commonly performed by placement of the wafer on a heated plate for a given period of time. The heated plate is of large thermal mass relative to the wafer and is held at a constant temperature by a feedback controller that adjusts the resistive heater power in response to a temperature sensor embedded in the plate near the surface. The plate is designed with multiple radial zone configurations. The wafer may be placed in direct contact or on proximity pins. A general requirement for these systems is the ability to reject the load disturbance induced by placement of a cold wafer on the bake-plate.

4.2 Physical Modeling

The physical model of a m -zone bake-plate as shown in Figure 4.2 has been derived [56]. According to [56], a distributed lumped model can satisfactorily describe the bake-plate characteristics, and with energy balance and heat transfer laws, a m -zone bake-plate can be modeled as

$$\begin{aligned}
 C_1 \dot{T}_1(t) &= P_1(t) - \frac{T_1(t)}{R_1} + \frac{T_2(t) - T_1(t)}{R_{1,2}}, \\
 C_i \dot{T}_i(t) &= P_i(t) + \frac{T_{i-1}(t) - T_i(t)}{R_{i-1,i}} - \frac{T_i(t)}{R_i} + \frac{T_{i+1}(t) - T_i(t)}{R_{i,i+1}}, \quad (i = 2, \dots, m-1) \\
 C_m \dot{T}_m(t) &= P_m(t) - \frac{T_m(t)}{R_m} + \frac{T_{m-1}(t) - T_m(t)}{R_{m-1,m}}.
 \end{aligned} \tag{4.1}$$

Figure 4.2: (a) Side view of bake plate; (b) Top view of bake plate

where

C_i = heat capacity for zone i

$T_i(t)$ = temperature of zone i

P_i = input heat power for zone i

R_i = thermal convection resistance between bakeplate zone i and surrounding air

$R_{i-1,i}$ = thermal convection resistance between bakeplate zone $i - 1$ and zone i

$R_{i,i+1}$ = thermal convection resistance between bakeplate zone i and zone $i + 1$

To write (4.1) in a more compact form, we have

$$\begin{aligned}
 C_1 \dot{T}_1(t) &= P_1(t) - \frac{T_1(t)}{R_{p1}} + \frac{T_2(t)}{R_{1,2}}, \\
 C_i \dot{T}_i(t) &= P_i(t) + \frac{T_{i-1}(t)}{R_{i-1,i}} - \frac{T_i(t)}{R_{pi}} + \frac{T_{i+1}(t)}{R_{i,i+1}} \quad (i = 2, \dots, m-1) \\
 C_m \dot{T}_m(t) &= P_m(t) - \frac{T_m(t)}{R_{pN}} + \frac{T_{m-1}(t)}{R_{m-1,m}}
 \end{aligned} \tag{4.2}$$

where R_{pi} is the overall thermal convection resistance of zone i and can be calculated as

$$\begin{aligned}
 \frac{1}{R_{p1}} &= \frac{1}{R_1} + \frac{1}{R_{1,2}} \\
 \frac{1}{R_{pi}} &= \frac{1}{R_i} + \frac{1}{R_{i-1,i}} + \frac{1}{R_{i,i+1}}, \quad i = 2, \dots, m-1 \\
 \frac{1}{R_{pN}} &= \frac{1}{R_m} + \frac{1}{R_{m-1,m}}
 \end{aligned} \tag{4.3}$$

At steady state, $\dot{T}_i(\infty) = 0$ and (4.2) becomes

$$\begin{aligned}
 P_1(\infty) &= \frac{T_1(\infty)}{R_{p1}} - \frac{T_2(\infty)}{R_{1,2}} \\
 P_i(\infty) &= -\frac{T_{i-1}(\infty)}{R_{i-1,i}} + \frac{T_i(\infty)}{R_{pi}} - \frac{T_{i+1}(\infty)}{R_{i,i+1}} \\
 P_m(\infty) &= \frac{T_m(\infty)}{R_{pN}} - \frac{T_{m-1}(\infty)}{R_{m-1,m}}
 \end{aligned} \tag{4.4}$$

Since the baking process will be conducted around a constant temperature, it will be more convenient if deviation variables are used in the plant model. Let $\theta_i(t)$ be the temperature deviation of zone i from the steady state, and $u_i(t)$ be the input

power deviation of zone i from the steady state, i.e.

$$\begin{aligned}\theta_i(t) &= T_i(t) - T_i(\infty) \\ u_i(t) &= P_i(t) - P_i(\infty)\end{aligned}\tag{4.5}$$

Then

$$\begin{aligned}T_i(t) &= \theta_i(t) + T_i(\infty) \\ P_i(t) &= u_i(t) + P_i(\infty) \\ \dot{\theta}_i(t) &= \dot{T}_i(t) \\ \dot{u}_i(t) &= \dot{P}_i(t)\end{aligned}\tag{4.6}$$

Substitute (4.3), (4.4) and (4.6) into equation (4.2), then we have

$$\begin{aligned}C_1 \dot{\theta}_1(t) &= u_1(t) + \frac{\theta_2(t)}{R_{1,2}} - \frac{\theta_1}{R_{p1}} \\ C_i \dot{\theta}_i(t) &= u_i(t) + \frac{\theta_{i-1}(t)}{R_{i-1,i}} + \frac{\theta_{i+1}(t)}{R_{i,i+1}} - \frac{\theta_i}{R_{pi}} \\ C_m \dot{\theta}_m(t) &= u_m(t) + \frac{\theta_{m-1}(t)}{R_{m-1,m}} - \frac{\theta_m}{R_{pN}}\end{aligned}\tag{4.7}$$

From (4.7) we can derive the state space model for a general m -zone bake plate in continuous time as

$$\begin{aligned}\dot{x} &= A_c x + B_c u \\ y &= x\end{aligned}\tag{4.8}$$

with

$$\begin{aligned} x &= \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_m \end{bmatrix}^T \\ u &= \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}^T \\ y &= \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_m \end{bmatrix}^T \end{aligned}$$

$$A_c = \begin{bmatrix} \frac{-1}{C_1 R_{p1}} & \frac{1}{C_1 R_{1,2}} & & & 0 \\ \frac{1}{C_2 R_{1,2}} & \frac{-1}{C_2 R_{p2}} & \frac{1}{C_2 R_{2,3}} & & \\ & \frac{1}{C_3 R_{2,3}} & \frac{-1}{C_3 R_{p3}} & \ddots & \vdots \\ & & \ddots & \ddots & \frac{1}{C_{m-1} R_{m-1,m}} \\ 0 & & & \frac{1}{C_N R_{m-1,m}} & \frac{-1}{C_N R_{pN}} \end{bmatrix}$$

and

$$B_c = \begin{bmatrix} \frac{1}{C_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{C_m} \end{bmatrix}$$

In this application, a three-zone bake plate (see Figure 4.3 for photograph) is used.

Apply the general m -zone model to this specific application, we then obtain

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_1 R_{p1}} & \frac{1}{C_1 R_{1,2}} & 0 \\ \frac{1}{C_2 R_{1,2}} & \frac{-1}{C_2 R_{p2}} & \frac{1}{C_2 R_{2,3}} \\ 0 & \frac{1}{C_3 R_{2,3}} & \frac{-1}{C_3 R_{p3}} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & \frac{1}{C_3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (4.9)$$

Figure 4.3: Photograph of the bake plate.

with

$$\begin{aligned}\frac{1}{R_{p1}} &= \frac{1}{R_1} + \frac{1}{R_{1,2}} \\ \frac{1}{R_{p2}} &= \frac{1}{R_2} + \frac{1}{R_{1,2}} + \frac{1}{R_{2,3}} \\ \frac{1}{R_{p3}} &= \frac{1}{R_3} + \frac{1}{R_{2,3}}\end{aligned}$$

Most of the parameters can be obtained or calculated from the thermodynamics handbook, for convenience, we have listed the relevant parameters in Table 4.1.

Table 4.1: The material characteristic

Parameters	Value	Unit
h (natural convection coefficient of air)	24	W/m.K
C_{al} (specific heat capacity of aluminium)	917	J/kg.K
ρ_{al} (density of aluminium)	2700	kg/m ³
tp (Plate thickness)	0.021	m
Δr (width of each zone)	0.03	m

The thermal capacitances and resistances are given by:

$$\begin{aligned}
 C_i &= C_{al} \times \rho_{al} \times tp \times A_i, \\
 R_i &= \frac{1}{hA_i}, \\
 A_i &= \pi \Delta r [i^2 - (i-1)^2], \quad i = 1, 2 \text{ and } 3.
 \end{aligned}$$

where A_i is the cross-sectional area of zone i . Finally, we can calculate the values of the various thermal capacitances and resistance as follows:

$$\begin{aligned}
 C_1 &= 147 \text{ J/K}, \quad C_2 = 441.02 \text{ J/K}, \quad C_3 = 735.04 \text{ J/K} \\
 R_1 &= 14.73 \text{ K/W}, \quad R_2 = 4.91 \text{ K/W}, \quad R_3 = 2.94 \text{ K/W}
 \end{aligned}$$

The remaining unknown parameters in (4.9) are the thermal resistances $R_{1,2}$ and $R_{2,3}$ due to the heat transfer between the zones. They can be estimated through experiments.

4.3 Parameter Estimation of the Three-zone Bake Plate

Although most of the parameters in the model developed in the previous section can be obtained from handbooks, to obtain a more accurate and realistic model of the actual system, actual input-output data are necessary to estimate the various parameters in (4.9). We will use the open-loop test to do the parameter estimation because it is simple. In the following, we will first outline the experimental setup and then use the experimental data to estimate a model of the three-zone bake plate.

4.3.1 Experimental Setup

The experimental setup for the parameter estimation as well as the temperature control is shown in Figure 4.4 with the photograph shown in Figure 4.5. A control software system was developed using the National Instruments LabVIEW programming environment to create a dynamic temperature control system. SCR (Silicon Controlled Rectifier) type power controller is used to control the input power to the heaters in each zone. The input dc voltage of the power controller ranging from 0 to 5V and is proportional with the input power to the heater. The bake plate in this application is divided into three zones with each zone having its own embedded heaters and temperature sensors (see Figure 4.2 for the configuration of the bake plate). The sensors will measure the plate temperature and the value will be sent back to the PC through signal conditioning instrumentation.

4.3.2 Parameter Estimation

The open-loop tests are conducted following the procedure below:

- (1) Bring the temperature of all zones to a steady state around 90°C (since

Figure 4.4: Experimental setup diagram

Figure 4.5: Photograph of the experimental setup.

conventional soft bake usually works at this temperature [57], record the steady state input power and the temperature for the three zones.

- (2) Apply a step input voltage of magnitude $0.3V$ (i.e. a step input power to heater) on zone 1 while maintain the input power of heaters on the other two zones unchanged. Record the temperature change from the steady state for all the three zones.
- (3) Repeat Step 2 for the other two zones.

In this application, a low order model of the system suffice, hence a step test is employed. Seven thousand data points were collected with sampling period of one second.

Since the physical model derived in Section 4.2 suggests a particular structure for the state space matrices (see Eq.(4.9)), which includes coupled unknown parameters, we could use “grey-box” identification instead of “black-box” identification. Roughly speaking, the plant model can be represented by

$$\begin{aligned}\dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) \\ y(t) &= C(\theta)x(t)\end{aligned}$$

where θ is the vector of unknown parameters to be estimated by minimizing the cost function

$$V_N = \sum_{t=1}^N \varepsilon(t, \theta)^T \varepsilon(t, \theta)$$

and $\varepsilon(t, \theta)$ is the difference between the measured output and the predicted output of the model. We used Matlab System Identification Toolbox, in particular the “idgrey” command to define the plant model structure Eq.(4.9) and “pem” command to carry out the parameter estimation [58, 59].

This method needs the nominal (initial) values of the parameters, which we have already known in Section 4.2. Using the parameters as initial values, the estimated parameter values are obtained as follows:

$$\begin{aligned}C_1 &= 190.21 J/K, & C_2 &= 515.039 J/K, & C_3 &= 557.848 J/K \\ R_1 &= 8.158 K/W, & R_2 &= 3.1207 K/W, & R_3 &= 1.274 K/W \\ R_{12} &= 2.754 K/W, & R_{23} &= 0.583 K/W\end{aligned}$$

Substitute the estimated parameters to (4.9), we have the state-space model of the three zone bake plate in continuous-time:

$$\begin{aligned}\dot{x} &= A_c x + B_c u \\ y &= x\end{aligned}\tag{4.10}$$

where

$$A_c = \begin{bmatrix} -0.0025525 & 0.0019085 & 0 \\ 0.0007048 & -0.004652 & 0.003324 \\ 0 & 0.0030697 & -0.004471 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0.0052573 & 0 & 0 \\ 0 & 0.0019416 & 0 \\ 0 & 0 & 0.0017926 \end{bmatrix}$$

4.3.3 Model Validation

The models described in the previous sections are derived either based on the physical understanding of the thermal modeling (physical model) or parameter estimations (estimated model) based on the input-output data collected by conducting a set of open-loop tests. The results of the model fitting and validation with the data from the open-loop tests are shown in Figure 4.6. The figure illustrate the close match obtained between the estimated model (4.10) and the actual process, which makes it reasonable to be used in a model-based controller design.

From Figure 4.6, we also observe clear mismatch of outputs between the physical model and the actual process. This result tell us that the physical model is helpful in deciding the model structure and also as initial value for parameter estimation. However, it is not suitable for controller design.

Another observation from Figure 4.6 is that the interactions between the three zones are not negligible. Therefore, it is better to employ multivariable control techniques rather than to use three single-loop controllers.

Figure 4.6: Model verification using step response. from top to bottom, step input applied at zone-1, zone-2 and zone-3.

4.4 Experimental Results

Figure 4.7 shows the experimental result without constraints. It could be seen that the closed-loop system controlled by MMPC has faster disturbance rejection response than that controlled by traditional SMPC. It takes around 200s for MMPC to reject the disturbance and return to the set-point while for SMPC it takes about 300s. In this application, this means that the next wafer can be processed after 200s instead of 300s using MMPC. This translates to a 30% increase in the number

of wafer produced within a given time period. Figure 4.8 show the experimental result with input constraints. However, the performance improvement of MMPC over SMPC is not so apparent. This is because the achievable control performance is limited by the available control authority. When input constraints are active, both the MMPC and SMPC schemes employ similar control actions and the control signals are at the constraints most of the time, thus lead to very similar closed-loop performance. However, the on-line computation complexity involved in MMPC has been greatly reduced compared with that of SMPC, since in SMPC there are 15 ($N_u = 5$, $m = 3$) variables needed to be optimized every sampling time while in MMPC it is only 5, reduced by $1/3$.

In the experimental runs, the SMPC measured the temperature at 1.2s interval (at $t = 340.4s$, $341.6s$, $342.8s, \dots$ as marked by 'o' in Figure 4.9) while the MMPC measured the temperatures 3 times faster ($m = 3$) at 0.4s interval (at $t = 340.4s$, $340.8s$, $341.2s, \dots$ as marked by 'x' in Figure 4.9). As a result, the MMPC acquired more information than SMPC – a key reason for its better performance. In both MMPC and SMPC, the zero-order-hold for the control signal were 1.2s, the control horizons were $N_u = 5$ and the weighting matrices were $q = \text{diag}\{0.01, 0.1, 0.2\}$ and $r = 1$. As can be seen in Figure 4.9, the SMPC updates the three control signals simultaneously at interval of 1.2s, i.e., at $t = 340.4s$, $341.6s$ and $342.8s$ as indicated by the blue lines in the figure. In comparison, MMPC also updates the control signal at 1.2s interval, but in a multiplex fashion, i.e., control signal 1 at $t = 340.4s$, $341.6s$, $342.8s$; control signal 2 at $t = 340.8s$, $342.0s$ and $343.2s$; control signal 3 at $t = 341.2s$, $342.4s$ and $343.6s$.

One may be curious about why the comparison between SMPC and MMPC is not done with the same prediction horizon and the same sampling rate. The reason is as follows: Given the same prediction horizon say, $N = 5$ with the same sampling period T , N_u^s (N_u for SMPC) and N_u^m (N_u for MMPC) are

$$N_u^s = N = 5, \quad N_u^m = \frac{N-1}{m} + 1 = 2$$

In this case, SMPC should outperform MMPC since SMPC could execute 3 control actions simultaneously while MMPC can only execute one at a time. However, it should be noted that within one sampling period T , SMPC needs to solve a control problem with complexity of $O((15)^3)$, while MMPC only needs to solve a control problem with complexity of $O((2)^3)$. We reason that if one could implement SMPC in T with the available computing resources, then one could employ MMPC at a faster sampling rate of say $T/3$ due to its much reduced computational burden and it is logical that one would want to make good use of the available computing resources. This leads us back to the scenario of MMPC sampling 3 times more frequent than SMPC.

4.5 Computational Scalability of MMPC

To further explore the issue of computational scalability of MMPC, additional experiments were conducted. Figures 4.10 to 4.12 show the performance as well as the computational time with varying horizon length for both SMPC and MMPC. The computational time reported here were obtained by timing the quadratic optimization function calls during the course of each experiment. It could be observed from Figures 4.10 to 4.12 that when the constraints are active, the computational time of MMPC is much shorter than that of SMPC.

Table 4.2: Maximum Computational Time of MMPC and SMPC with Varying Horizon

Control Horizon (N_u)	Maximum Computational Time (sec)		
	MMPC (one control update)	MMPC (three control updates)	SMPC (three control updates)
5	0.02	0.06	0.05
20	0.10	0.30	0.74
25	0.15	0.45	1.29

As MMPC updates the 3 control signals sequentially, the computational time of one complete MMPC control cycle is compared with one SMPC control cycle. Table 4.2 shows the computational time needed for MMPC to update one control signal as well as for MMPC and SMPC to update three control signals. The numerical values in Table 4.2 are plotted in Figure 4.14 for a visual comparison. It could be observed that when $N_u = 5$, the computational time of MMPC in one control cycle is slightly longer than that of SMPC, while it is much shorter as the horizon increases. We hypothesize that this is due to overheads in the QP solver, such as set-up time, which dominate the solution time for small problems and therefore penalize the more frequent optimization calls of MMPC. However, as the horizon length increases, the computational time become dominated by the actual solution process and MMPC scales more favorably than SMPC.

Thus, for simple applications where the number of zones is small, or when N_u is small, SMPC may suffice. However, for more complex applications where the number of zones is large (the state of the art bake-plate has 49 zones) or when N_u is large, MMPC is computationally more scalable than SMPC.

4.6 Sampling Rate and Disturbance Rejection Performance

The reduction in computational load of MMPC can be gainfully used to increase sampling rate and improve performance. Figure 4.14 shows the experimental results of MMPC with two different sampling intervals, namely, 0.4s and 1.0s. The simulation results are also superimposed onto Figure 4.14 to show the match between the experimental and simulation results. As could be observed from Figure 4.14, a shorter sampling time can result in faster recovery of bake-plate temperature when a cold wafer is placed on the bake-plate. This result is important for the semiconductor wafer baking process, because temperature non-uniformity will

affect critical dimension of the wafer. Table 4.3 shows the integral-square-error of the temperature fluctuation computed from the data collected experimentally and from simulations. It is clear from Table 4.3 and Figure 4.14 that experimental and simulation results matched.

Table 4.3: Experimental and simulation results for different sampling period

Sampling Interval	Integral-Square-Error	
	Simulation ($\times 10^3$)	Experiment ($\times 10^3$)
0.4s	0.74	0.77
1.0s	1.23	1.16

4.7 Conclusion

In this chapter, the proposed MMPC was applied to an industrial application, which is the baking process for silicon wafer manufacturing. The physical modeling, as well as parameter estimation for a three zone bake plate, were described in detail followed by experimental implementation and evaluation of SMPC and MMPC. The experimental results showed that MMPC is a better control scheme than SMPC in rejecting the disturbance caused by the placement of a cold wafer on the hot plate. This result is important for the semiconductor wafer baking process, because temperature non-uniformity will affect critical dimension of the wafer. With control horizon $N_u = 5$, conventional MPC needs to optimize $m \times N_u = 3 \times 5 = 15$ variables every sampling time whereas MMPC only needs to optimize 5 variables. This computational advantage of MMPC becomes even more significant when constraints are considered and with increasing number of zones and control horizon. For example, consider the state-of-the-art 49-zone bake-plate [1] with $N_u = 5$. SMPC would have to solve a Quadratic Program (QP) of $49 \times 5 = 245$ variables which could result in a high dimension on-line optimization challenge, whereas MMPC would solve 49 QP of only 5 variables each. The computational time of

MMPC and SMPC have been compared. It has been demonstrated that MMPC scales better than SMPC. This result is important especially when considering applying MPC to state-of-art bake plate which has 49 zones. In addition, the computational time savings achieved by MMPC could be gainfully used to increase the sampling rate, which leads to faster disturbance rejection. Faster disturbance rejection could lead to increased throughput in wafer production process because the next wafer can be processed earlier. Figure 4.14 showed that the temperature of the bake-plate recovered back to 90°C at $t = 350\text{s}$ and $t = 480\text{s}$ when the sampling time was 0.4s and 1.0s respectively. Hence, when the longer sampling time was used, if the next wafer comes at $t = 350\text{s}$, then the temperature of the bake plate would not have recovered back to 90°C and the second wafer would be baked at a different starting temperature. When this happens, there will be wafer-to-wafer non-repeatability which will affect critical dimension of the wafer. It is suggested that MMPC could serve as a promising alternative for this class of applications.

Figure 4.7: Disturbance rejection performance comparison of MMPC and SMPC (Experimental result without constraints).

Figure 4.8: Disturbance rejection performance comparison of MMPC and SMPC (Experimental result with constraints).

Figure 4.9: Pattern of input update for conventional synchronized MPC('bo') and for the Multiplexed MPC('r*').

Figure 4.10: Performance and computational time of MMPC and SMPC with $N_u = 5$

Figure 4.11: Performance and computational time of MMPC and SMPC with $N_u = 20$

Figure 4.12: Performance and computational time of MMPC and SMPC with $N_u = 25$

Figure 4.13: Plot of SMPC and MMPC maximum computational time vs control horizon.

Figure 4.14: Temperature responses of MMPC with sampling interval 0.4s (Solid Red) and 1.0s (Dashed-dot Red) superimposed with simulation (Dashed Blue)

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have investigated methods to make MPC scalable and applicable to embedded applications or systems with a large number of inputs. We proposed a novel approach, Multiplexed MPC, to solve this problem. The computational complexity of typical MPC problems is $O((mN_u)^3)$, where m is the number of control inputs and N_u is the control horizon. In contrast, the MMPC scheme solves m control problems, each having a computational complexity $O((N_u)^3)$. Thus, the total computation for MMPC is $m \times O((N_u)^3)$ and this compares favorably with standard MPC's $O((mN_u)^3)$ especially when N_u is large. In addition, in this thesis, we investigated the stability and performance issues of MMPC. A more detailed summary is given below:

- We have addressed the problem of implementing embedded constrained MPC algorithm for multivariable system and proposed a novel control scheme called multiplexed MPC.
- It has been shown that by a properly chosen terminal cost and ensuring that the problem is well posed, the nominal stability of the proposed MMPC

scheme can be established. The computation and performance study show that MMPC is effective in that it reduced the computational complexity while at no worsen control performance.

- We also showed that when active constraints set is empty or known, MMPC can be interpreted as a piecewise linear periodic state feedback law in the augmented state space. This provides a framework for systematic analysis and investigation of the properties of MMPC. Based on the augmented state space, the monodromy matrix for the MMPC controlled system could be systematically worked out and analyzed.
- The formula for computing the quadratic cost of MMPC is derived and illustrated in the augmented state space framework.
- We have applied the cost formula to evaluate the effects of the design choices available in MMPC on the closed-loop performance. This could help the MMPC designer to choose suitable tuning parameters and also the input updating patterns.
- Finally, the proposed Multiplexed MPC scheme is successfully applied to a multi-zone heating process typically encountered in semiconductor manufacturing. The experimental results showed that Multiplexed MPC indeed is computationally more scalable than conventional MPC. This reduction in computational load has been gainfully used to increase the sampling rate of MMPC, leading to better disturbance rejection performance. This result is important for semiconductor wafer baking process as it could lead to increased throughput as well as reduction in wafer-to-wafer non-repeatability which will affect critical dimension of the wafer.

5.2 Recommendations for Future work

Multiplexed MPC is a new technology especially applicable for embedded application. This thesis has made some initial research on this novel technique, establishing nominal stability result and laid a foundation for further research. In this section, we recommend some ideas for future investigation:

- Development of MMPC to systems with more general updating patterns. For example, the control inputs can be grouped into subsets with each updated with different frequencies. On the other hand, it is also interesting to investigate the interval between two updating instances, to what extent could it be squeezed to approximate the synchronized MPC? How to decide a proper updating interval to get a better compromise between computation complexity and control performance?
- Robustness analysis of the MMPC in the presence of model uncertainty and disturbance. We have proved the nominal stability of the proposed MMPC scheme in Chapter 2. Since modeling error always exists in practice and disturbances are inevitable, it is necessary to ensure that the resulting closed-loop system is stable in the presence of modeling error and disturbance. Robust MPC has received much attention and remains an active area for research. A recent survey [23] gives a thorough list of work in this area so far. [60] considers robustness analysis of nominal MPC formulations and [61] presented some robustness results of receding horizon control with LMI formulations. In addition, [62, 63] proposed a new RMPC method which aimed at constrained system with time-varying model uncertainty with various uncertainty set. A key challenge in designing a robust MPC algorithm suitable for embedded applications is to achieve the desired robustness without making the resulting computation complex. A recent work [64] extended the multiplexed MMPC with guaranteed robustness under uncertain but bounded disturbance. However, it is a very initial study, more thorough work still

needs to be done.

- Application of the developed MMPC algorithm to the practical embedded system. To verify the effectiveness, experiments should be carried out on different real-time processes. Analysis on the computation time and resources utilization of MMPC compared with that of SMPC may be conducted based on the embedded application. At present, study of embedded MPC could be found in the literature, such as [20, 41, 42, 43, 44, 45, 65]. They are concerned with embedded MPC algorithm through either FPGA or micro processor and have obtained some initial results on computation speed and resources usage such as memory and ALUs (Arithmetic Logic Units). By employing MMPC algorithm to the embedded system with multi-inputs and outputs, the computation speed could be greatly increased and the computation complexity could be further reduced by exploiting the specific structure which may exist in specific applications.

Author's Publications

Publications related to this thesis:

- [1] K.-V. Ling, W.-K. Ho, **B.-F. Wu**, Andreas Lo and H. Yan, “Multiplexed MPC for multi-zone thermal processing in semiconductor manufacturing”, Accepted for publication in *IEEE Transaction on Control Systems Technology*.
- [2] K.-V. Ling, W.-K. Ho, **B.-F. Wu**, A. G. Aribowo, Y. Feng and H. Yan, “Experimental Evaluation of Multiplexed MPC for semiconductor manufacturing”, *7th Asian Control Conference*, Hong Kong, China, August 2009.
- [3] K.-V. Ling, J.-M. Maciejowski and **B.-F. Wu**, “Computing the Cost in Multiplexed MPC”, *10th ICARCV World Conference*, Hanoi, Vietnam, December 2008.
- [4] K.-V. Ling, J.-M. Maciejowski and **B.-F. Wu**, “Multiplexed Model Predictive Control”, *CUED Report*, CUED/F-INFENG/TR.561, July 2006.
- [5] K.-V. Ling, J.-M. Maciejowski and **B.-F. Wu**, “Multiplexed Model Predictive Control”, *16th IFAC World Congress*, Prague, July 2005.

Others:

- [1] K.-V. Ling, **B.-F. Wu** and J.-M. Maciejowski, “Embedded Model Predictive Control (MPC) using a FPGA” *19th IFAC World Congress*, Seoul, July 2008.
- [2] K.-V. Ling, M.-H. He, **B.-F. Wu** and Y. Zhang, “An Augmented Model Predictive Controller for Multirate Cascade Systems”, *American Control Conference*,

5.2 Recommendations for Future work

87

Boston, USA, pp.1575-1579, 30 June - 2 July 2004.

Bibliography

- [1] K. El-Awady, C. Schapter, and T. Kailath, “Programmable thermal processing module for semiconductor substrates,” *IEEE Transactions on Control System Technology*, vol. 12, pp. 493–509, 2004.
- [2] W. H. Kwon and D. G. Byun, “Receding horizon tracking control as a predictive control and its stability properties,” *International Journal of Control*, vol. 50, pp. 1807–1824, 1989.
- [3] W. H. Kwon and S. Han, *Receding Horizon Control : Model Predictive Control for State Models*. Springer, 2005.
- [4] C. E. García, D. Prett, and M. Morari, “Model predictive control: theory and practice-a survey,” *Automatica*, vol. 25, pp. 335–348, 1989.
- [5] J. Richalet, “Industrial application of model based predictive control,” *Automatica*, vol. 29, pp. 1251–1274, 1993.
- [6] S. J. Qin and T. A. Badgell, “A survey of industrial model predictive control technology,” *Chemical Engineering Progress*, vol. 11, pp. 733–764, 2003.
- [7] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. New York: Springer, 2004.
- [8] M. Morari and J. H. Lee, “Model predictive control: past, present, and future,” *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.

- [9] J. M. Maciejowski, *Predictive Control with Constraints*. Harlow UK: Prentice-Hall, 2002.
- [10] Y. Nesterov and A. Nemirovskii, “Interior point polynomial algorithms in convex programming,” in *Studies in applied mathematics (Vol. 13)*, Philadelphia, PA: SIAM, 1994.
- [11] M. Cannon, B. Kouvaritakis, and J. A. Rossiter, “Efficient active set optimization in triple mode MPC,” *IEEE Transactions on Automatic Control*, vol. 46, pp. 1307–1312, 2001.
- [12] S. J. Wright, “Application of interior-point method to model predictive control,” *Journal of Optimization Theory and Applications*, vol. 99, no. 3, 1998.
- [13] T. Perez, G. Goodwin, and C. Tzeng, “Model predictive rudder roll stabilization control for ships,” in *Proc. 5th IFAC Conf. on Manoeuvring and Control of Marine Craft*, Aalborg, Denmark, Jan 2000.
- [14] P. Kaczynski and P. Domanski, “Evolutionary predictive controller for ship control,” in *ICSC Congress on Computational Intelligence Methods and Applications*, Istanbul, Turkey, 2005.
- [15] R. Murray, J. Hauser, A. Jadabie, M. Milam, N. Petit, W. Dunbar, and R. Franz, “Online control customization via optimization-based control,” in *Software-Enabled Control*, T. Samad and G. Balas, Eds. IEEE Press and Wiley, 2003.
- [16] I. Skrjanc, “A decomposed-model predictive functional control approach to air-vehicle pitch-angle control,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 48, no. 1, pp. 115–127, 2007.
- [17] A. Richards and J. How., “Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,” in *Proc. American Control Conference*, Denver, June 2003.

- [18] M. Morari, M. Baotić, and F. Borrelli, “Hybrid systems modeling and control,” *European Journal of Control*, vol. 9, no. 2-3, pp. 177–189, 2003.
- [19] B. Kim, D. Neculescu, and J. Sasiadek, “Model predictive control of an autonomous vehicle,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, Piscataway, NJ, USA, 2001.
- [20] L. G. Bleris and M. V. Kothare, “Real-time implementation of model predictive control,” in *American Control Conference*, Portland, OR, USA, June 2005.
- [21] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*. London: CRC press, 2003.
- [22] G. C. Goodwin, M. Seron, and J. D. Dona, *Constrained Control and Estimation. An Optimization Approach*. Springer-Verlag, 2005.
- [23] A. Bemporad and M. Morari, “Robust model predictive control: a survey,” *Robustness in Identification and Control*, vol. 245, pp. 207–226, 1999.
- [24] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: stability and optimality,” *Automatica*, vol. 36, pp. 789–814, 2000.
- [25] A. Grancharova and T. A. Johansen, “Survey of explicit approaches to constrained optimal control,” *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 47–97, 2005.
- [26] R. R. Negenborn, B. D. Schutter, and J. Hellendoorn, “Multi-agent model predictive control: A survey,” Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, Tech. Rep. 04-010, 2004.
- [27] The Mathworks, “Model Predictive Control Toolbox ,” 2009. [Online]. Available: <http://www.mathworks.com/products/mpc/>

- [28] M. Kvasnica, P. Grieder, and M. Baotić, “Multi-Parametric Toolbox (MPT),” 2004. [Online]. Available: <http://control.ee.ethz.ch/~mpt/>
- [29] National Instrument, “LabVIEW Control Design User Manual,” 2009. [Online]. Available: <http://www.ni.com/labview/>
- [30] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, “Distributed model predictive control,” *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.
- [31] D. Jia and B. Krogh, “Distributed model predictive control,” in *Proc. American Control Conference*, Arlington, VA, 2001.
- [32] D. Jia and B. H. Krogh, “Min-max feedback model predictive control for distributed control with communication,” in *Proceedings of the American Control Conference*, Anchorage, Alaska, May 2002.
- [33] F. B. T. Keviczky and G. J. Balas., “A study on decentralized receding horizon control for decoupled systems,” in *Proceedings of the American Control Conference*, Boston, Massachusetts, July 2004.
- [34] A. Richards and J. How., “A decentralized algorithm for robust constrained model predictive control,” in *Proc. American Control Conference*, Massachusetts, June 2004.
- [35] A. Venkat, J. Rawlings, and S. Wright, “Plant-wide optimal control with decentralized MPC,” in *Proc. 7th International Symposium on Dynamics and Control of Process Systems (DYCOPS)*, Cambridge, MA, July 2004.
- [36] M.-M. Ben Gaid, A. Çela, and Y. Hamam, “Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system,” *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, July 2006.

- [37] K. V. Ling, J. M. Maciejowski, and B. F. Wu., “Multiplexed model predictive control,” in *16th IFAC World Congress*, Prague, July 2005.
- [38] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed. Philadelphia PA: SIAM, 1994.
- [39] J. A. Rossiter, J. Sheng, T. Chen, and S. L. Shah, “Interpretations of and options in dual-rate predictive control,” *Journal of Process Control*, vol. 15, pp. 135–148, 2005.
- [40] 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.
- [41] J. M. G. Arnold, L. G. Bleris, and M. V. Kothare, “LNS architectures for embedded model predictive control processors,” in *Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*, Washington DC, USA, September 2004.
- [42] L. G. Bleris, J. Garcia, M. V. Kothare, and M. G. Arnold, “Towards embedded model predictive control for system-on-a-chip applications,” *Journal of Process Control*, vol. 16, pp. 255–264, 2006.
- [43] L. G. Bleris, P. Vouzis, M. G. Arnold, and M. V. Kothare, “A co-processor FPGA platform for the implementation of real-time model predictive control,” in *American Control Conference*, Minneapolis, Minnesota, June 2006.
- [44] G. Hassapis, “Implementation of model predictive control using real-time multiprocessing computing,” *Microprocessors and Microsystems*, vol. 27, pp. 327–340, 2003.

- [45] K. V. Ling, S. P. Yue, and J. M. Maciejowski, "A FPGA implementation of model predictive control," in *American Control Conference*, Minneapolis, MN, June 2006.
- [46] Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [47] H. Richter and A. Singaraju, "Multiplexed predictive control of a large commercial turbofan engine," *Journal of Guidance, Control and Dynamics*, vol. 31, 2008.
- [48] S. Bittanti, P. Colaneri, and G. D. Nicolao, "The difference periodic Riccati equation for the periodic prediction problem," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 706–712, 1988.
- [49] S. Bittanti and P. Colaneri, "Invariant representations of discrete-time periodic systems," *Automatica*, vol. 36, pp. 1777–1793, 2000.
- [50] S. Bittanti, "Deterministic and stochastic linear periodic systems," in *Time series and linear systems*. Berlin: Springer, 1986.
- [51] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transaction on Automatic Control*, vol. 36, pp. 1008–1020, 1991.
- [52] T. Edgar, S. Butler, W. Campbel, C. Pfeiffer, C. Bode, S. Hwang, K. Balakrishnan, and J. Hahn, "Automatic control in microelectronics manufacturing: practices, challenges, and possibilities," *Automatica*, vol. 36, no. 11, 2000.
- [53] W. K. Ho, A. Tay, and C. D. Charles, "Optimal predictive control with constraints for the processing of semiconductor wafers on bake plates," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 1, pp. 88–96, 2000.
- [54] A. Tay, W. K. Ho, C. D. Charles, and L. L. Lee, "Constraint feedforward control for thermal processing of quartz photomasks in microelectronics manufacturing," *Journal of Process Control*, vol. 14, pp. 31–39, 2004.

- [55] A. Tay, W. K. Ho, N. Hu, and Y. Zhou, "Fault detection and estimation of wafer warpage profile during thermal processing in microlithography," in *AICHE Annual Meeting*, Austin, Texas, November 2004.
- [56] W. K. Ho, A. Tay, M. Chen, and C. M. Kiew, "Optimal feed-forward control for multizone baking in microlithography," *Ind. Eng. Chem. Res.*, vol. 46, pp. 3623–3628, 2007.
- [57] B. W. Smith, *Resist processing Microlithography: Science and Technology*. J. Sheats and B.W. Smith, ed., 1997.
- [58] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, 1999.
- [59] The Mathworks, "System Identification Toolbox ," 2009. [Online]. Available: <http://www.mathworks.com/products/sysid/>
- [60] E. C. Kerrigan and J. M. Maciejowski, "Robust feasibility in model predictive control: necessary and sufficient conditions," in *40th IEEE CDC*, Orlando FL, December 2001.
- [61] J. Primbs and V. Nevistić, "A framework for robustness analysis of finite receding horizon control," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1828–1838, 2000.
- [62] Y. J. Wang and J. B. Rawlings, "A new robust model predictive control method I: Theory and computation," *Journal of Process Control*, vol. 14, no. 3, pp. 249–262, 2004.
- [63] Y. J. Wang, "Robust model predictive control," Ph.D. dissertation, University of Wisconsin-Madison, 2002.
- [64] A. G. Richards, K. V. Ling, and J. M. Maciejowski, "Robust multiplexed model predictive control," in *European Control Conference*, Kos, Greece, July 2007.

-
- [65] K. V. Ling, B. F. Wu, and J. M. Maciejowski, “Embedded model predictive control (MPC) using a FPGA,” in *19th IFAC World Congress*, Seoul, July 2008.
- [66] S. Bittanti, P. Colaneri, and G. D. Nicolao, “The periodic Riccati equation,” in *The Riccati Equation*, S. Bittanti, A. J. Laub, and J. C. Willems, Eds. Berlin: Springer-Verlag, 1991.
- [67] E. C. Kerrigan, “Robust constrained satisfaction: invariant sets and predictive control,” Ph.D. dissertation, University of Cambridge, 2000.

Appendix

A Derivation of the Stabilizing Terminal Weight for Linear Periodic Control Systems

In this section, we show how the terminal weight for linear periodic control system and a stabilizing feedback gain can be computed. The solution to the unconstrained infinite horizon periodic optimal control problem is well studied (see [66]). The optimal control is

$$\Delta \tilde{u}_k = -(B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A x_k = -K_{\sigma(k)} x_k \quad (\text{A1})$$

where $P_{(\cdot)}$ is the backward solution of the following discrete-time periodic Riccati equation (DPRE)

$$P_k = A^T P_{k+1} A - A^T P_{k+1} B_{\sigma(k)} (B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A + q \quad (\text{A2})$$

It has been pointed out that, in general, the solution $P_{(\cdot)}$ need not be periodic, unless a suitable final condition is chosen. Such a final condition is referred to as a periodic generator [66]. [48] discuss (mild) conditions for the existence and uniqueness of such a solution of (A2), and provide algorithms for finding it; note that this solution can be pre-computed off-line.

Here, we follow the time-invariant approach as suggested in [66]. We begin with the following infinite horizon control problem for the periodic system of (3.1). The cost function is

$$J = \sum_{i=0}^{\infty} \|x_{k+i+1}\|_q^2 + \|\Delta \tilde{u}_{k+i}\|_r^2 \quad (\text{A3})$$

This cost function can be re-written as

$$\begin{aligned} J &= \sum_{j=0}^{\infty} \sum_{i=1}^m \|x_{k+jm+i}\|_q^2 + \|\Delta \tilde{u}_{k+jm+i-1}\|_r^2 \\ &= \sum_{j=0}^{\infty} \|\bar{x}_{k+jm}\|_{\bar{q}}^2 + \|\Delta \bar{u}_{k+jm}\|_{\bar{r}}^2 \end{aligned}$$

where $\bar{q} = \text{diag}(q, \dots, q)$, $\bar{r} = \text{diag}(r, \dots, r)$,

$$\bar{x}_k = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+m} \end{bmatrix}, \quad \Delta \bar{u}_k = \begin{bmatrix} \Delta \tilde{u}_k \\ \Delta \tilde{u}_{k+1} \\ \vdots \\ \Delta \tilde{u}_{k+m-1} \end{bmatrix}$$

The “lifted” signals $\bar{x}_k$ and $\Delta \bar{u}_k$ can be constructed as

$$\begin{aligned} \bar{x}_k &= \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^m \end{bmatrix} x_k + \begin{bmatrix} B_{\sigma(k)} & \dots & \dots & 0 \\ AB_{\sigma(k)} & B_{\sigma(k)} & \ddots & \vdots \\ \vdots & & \ddots & \vdots \\ A^{m-1}B_{\sigma(k)} & \dots & \dots & B_{\sigma(k+m-1)} \end{bmatrix} \Delta \bar{u}_k \\ &= \bar{A}x_k + \bar{B}_{\sigma(k)}\Delta \bar{u}_k \end{aligned}$$

so that

$$\begin{aligned} \mathcal{J} &= \sum_{j=0}^{\infty} \|\bar{x}_{k+jm}\|_{\bar{q}}^2 + \|\Delta \bar{u}_{k+jm}\|_{\bar{r}}^2 \\ &= \sum_{j=0}^{\infty} \|\bar{A}x_{k+jm} + \bar{B}_{\sigma(k)}\Delta \bar{u}_{k+jm}\|_{\bar{q}}^2 + \|\Delta \bar{u}_{k+jm}\|_{\bar{r}}^2 \\ &= \sum_{j=0}^{\infty} \|x_{k+jm}\|_{\bar{A}^T \bar{q} \bar{A}}^2 + \|\Delta \bar{u}_{k+jm}\|_{\bar{B}_{\sigma(k)}^T \bar{q} \bar{B}_{\sigma(k)} + \bar{r}}^2 + 2x_{k+jm}^T \bar{A}^T \bar{q} \bar{B}_{\sigma(k)} \Delta \bar{u}_{k+jm} \end{aligned}$$

subject to $x_{k+m} = A^m x_k + \tilde{B}_{\sigma(k)} \Delta \bar{u}_k$ where

$$\tilde{B}_{\sigma(k)} = \begin{bmatrix} A^{m-1} B_{\sigma(k)}, & \dots & B_{\sigma(k+m-1)} \end{bmatrix}$$

This is a standard infinite horizon optimal control problem with a cross term.

Let

$$Q = \bar{A}^T \bar{q} \bar{A} - S^T R^{-1} S$$

$$R = \bar{B}_{\sigma(k)}^T \bar{q} \bar{B}_{\sigma(k)} + \bar{r}$$

$$S = \bar{A}^T \bar{q} \bar{B}_{\sigma(k)}$$

$$\tilde{A} = A^m - R^{-1} S^T$$

$$v_k = \Delta \bar{u}_k + R^{-1} S^T x_k$$

The optimal control problem is equivalent to minimising the following cost function

$$J = \sum_{j=0}^{\infty} \|x_{k+jm}\|_Q^2 + \|v_{k+jm}\|_R^2$$

subject to $x_{k+m} = \tilde{A} x_k + \tilde{B}_{\sigma(k)} v_k$.

The optimal control law is

$$\Delta \bar{u}_k = -(\tilde{K} + R^{-1} S^T) x_k = -\bar{K}_{\sigma(k)} x_k$$

where

$$\tilde{K} = (\tilde{B}_{\sigma(k)}^T P_{\sigma(k)} \tilde{B}_{\sigma(k)} + R)^{-1} \tilde{B}_{\sigma(k)}^T P_{\sigma(k)} \tilde{A}$$

and $P_{\sigma(k)}$ satisfies the following ARE (Algebraic Riccati Equation)

$$P_{\sigma(k)} = Q + \tilde{A}^T P_{\sigma(k)} \tilde{A} - \tilde{A}^T P_{\sigma(k)} \tilde{B}_{\sigma(k)} (\tilde{B}_{\sigma(k)}^T P_{\sigma(k)} \tilde{B}_{\sigma(k)} + R)^{-1} \tilde{B}_{\sigma(k)}^T P_{\sigma(k)} \tilde{A}$$

The infinite horizon optimal cost is

$$J_{\sigma(k)} = x_k^T P_{\sigma(k)} x_k \tag{A4}$$

Once the value of $\sigma(k)$ is chosen, the terminal weight can be calculated as shown above. [66] discussed conditions (Proposition 6.1 and Theorem 6.11) when the time-invariant re-formulation of a periodic control problem coincides with the SPPS solution.

B Derivation of $\mathcal{X}_I(K_{\sigma(k)})$

In our problem formulation of (2.7), we assume that the terminal constraints are satisfied, that is, $x_{k+N|k} \in \mathcal{X}_I(K_{\sigma(k)})$. The feasible and invariant terminal set $\mathcal{X}_I(K_{\sigma(k)})$ is very crucial in stability guarantee. The algorithm to compute the invariant set for LTI system has been documented in [3, 51, 67].

Given an unforced linear system

$$x_{k+1} = Ax_k$$

with the state constraint

$$\chi = \{x | \phi x \leq \psi\}$$

where $\phi \in \mathbb{R}^{p \times n}$ and $\psi \in \mathbb{R}^{p \times 1}$. The maximal output admissible set is defined as

$$\mathcal{O}_\infty(A, \phi, \psi) = \{x | \phi A^k x \leq \psi, \forall k = 0, 1, 2, \dots\}.$$

(1) Take $j = 0$.

(2) For $i = 1$ to p , solve the following linear programming problem:

$$J_i = \max \quad \phi^i A^{j+1} z \quad w.r.t \quad z \quad \text{subject to} \quad \phi A^k z \leq \psi, \quad \forall k = 0, 1, 2, \dots, j$$

where ϕ^i denotes the i th row of the matrix ϕ .

(3) If $J_i > \psi^i$ for $i = 1$ to p , take $j = j + 1$ and repeat the previous step.

(4) If $J_i \leq \psi^i$ for $i = 1$ to p , then $\mathcal{O}_\infty = \mathcal{O}_j$.

The invariant toolbox in [67] has functions to calculate the maximum admissible set. Using the following MATLAB command

$$[0inf, tstar, fd] = oinfset (A, [\phi \quad \psi], tmax)$$

one can obtain the $\mathcal{O}_\infty$. The function is developed based on the algorithm above. Since MMPC is a periodic control scheme, we need to convert the nominal closed system into an LTI system first.

$$\begin{aligned} x_{k+1} &= (A + B_{\sigma(k)}x_k) = \phi_{\sigma(k)}x_k \\ x_{k+2} &= (A + B_{\sigma(k+1)})x_{k+1} = \phi_{\sigma(k+1)}\phi_{\sigma(k)}x_k \\ &\vdots = \vdots \\ x_{k+m} &= \phi_{\sigma(k+m-1)} \cdots \phi_{\sigma(k)}x_k = \varphi_{\sigma(k)}x_k \end{aligned}$$

define

$$\bar{x}_k = \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+m-1} \end{bmatrix}, \quad \bar{x}_{k+m} = \begin{bmatrix} x_{k+m} \\ x_{k+m+1} \\ \vdots \\ x_{k+2m-1} \end{bmatrix}$$

Then

$$\bar{x}_{k+m} = \bar{\Psi}_{\sigma(k)}\bar{x}_k$$

$$\text{where } \bar{\Psi}_{\sigma(k)} = \begin{bmatrix} \varphi_{\sigma(k)} & & & \\ & \varphi_{\sigma(k+1)} & & \\ & & \ddots & \\ & & & \varphi_{\sigma(k+m-1)} \end{bmatrix}$$

Then the state constraints can be represented by

$$\Upsilon_{\sigma(k)}\bar{x}_k \leq \Omega_{\sigma(k)}$$

Let

$$\bar{\Psi}_{\sigma(k)} \rightarrow A, \quad \Upsilon_{\sigma(k)} \rightarrow \phi, \quad \Omega_{\sigma(k)} \rightarrow \psi$$

Then use the MATLAB command shown above, at time index k , one can obtain the invariant set of $\mathcal{O}_\infty$, which is a series of inequalities of the form $L\bar{x}_k \leq \Gamma$, since

$$\bar{x}_k = \begin{bmatrix} I \\ \phi_{\sigma(k)} \\ \vdots \\ \phi_{\sigma(k+m-2)} \cdots \phi_{\sigma(k)} \end{bmatrix} x_k = \Pi_{\sigma(k)} x_k$$

Then the invariant set is actually $L\Pi_{\sigma(k)}x_k \leq \Gamma$.