공학박사 학위논문

Optimality Enhancement in Move-blocked Model Predictive Control and Offset-free Model Predictive Control

이동블록 및 잔류편차 제거 모델예측제어 기법의 최적성 향상

2020년 2월

서울대학교 대학원
화학생물공학부
손 상 환

Abstract

Optimality Enhancement in Move-blocked Model Predictive Control and Offset-free Model Predictive Control

Sang Hwan Son
School of Chemical and Biological Engineering
The Graduate School
Seoul National University

Model predictive control (MPC) is a receding horizon control which derives finite-horizon optimal solution for current state online by solving an optimal control problem. MPC has had a tremendous impact on both industrial and control research areas. There are several outstanding issues in MPC. MPC has to solve the optimization problem within a sampling period so that the reduction of online computational complexity is a one of the main research subject in MPC. Another major issue is model-plant mismatch due to the model based predictive approach so that offset-free tracking schemes by compensating model-plant mismatch or unmeasured disturbance has been developed. In this thesis, we focused on the optimality performance of move blocking which fixes the decision variables over

arbitrary time intervals to reduce computational load for on-line optimization in MPC and disturbance estimator approach based offset-free MPC which is the most standardly used method to accomplish offset-free tracking in MPC. We improve the optimality performance of move blocked MPC in two ways. The first scheme provides a superior base sequence by linearly interpolating complementary base sequences, and the second scheme provides a proper time-varying blocking structure with semi-explicit approach. Moreover, we improve the optimality performance of offset-free MPC by exploiting learned model-plant mismatch compensating signal from estimated disturbance data. With the proposed schemes, we efficiently improve the optimality performance while guaranteeing the recursive feasibility and closed-loop stability.

Keywords: Model predictive control, input parameterization, move-blocking, model-plant mismatch, offset-free tracking
Student Number: 2016-30232

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Model predictive control (MPC) is a receding horizon control which dervies finite-horizon optimal solution for current state on-line by solving an optimal control problem [1]. MPC has become a standard method for decision making in various fields [2, 3, 4, 5, 6, 7]. The popularity of MPC is based on the fact that the resulting operating strategy respects all the details of system and problem, including constraints and interactions [8].

One of the most outstanding issues in MPC is the online computational load required to solve the optimization problem within a sampling period. Move blocking is an input parameterization scheme which fixes the decision variables over arbitrary time intervals, commonly referred as blocks, and it is widely implemented to model predictive control (MPC) to reduce computational load for on-line optimization. However, existing move blocking schemes have limitations in construction of base sequence and selection of blocking structure. First, move blocking strategies parameterize either the input sequence or offset from the base sequence of input, but existing move blocking schemes use a fixed base sequence only and do not fully exploit the valuable properties from various base sequences. Second, though selection of blocking structure has a significant effect

on the optimality of moved blocked MPC because the blocking position act as the search direction in input sequence space, but existing move blocked MPC schemes apply arbitrary time-invariant blocking structures without consideration of the optimality of blocking structure due to the difficulty of deriving proper time-varying blocking structure on-line.

Thus, we propose the interpolated solution based move blocking strategy which parameterizes the offset from the convex combination of two complementary base sequences — infinite-horizon linear quadratic regulator solution and shifted previous solution — and optimises the interpolation parameter as an additional decision variable in the optimal control problem to overcome the above-mentioned first limitation. This allows the controller to exploit the valuable properties from both solutions by choosing the optimal interpolation parameter and blocked offset according to the current state on-line while guaranteeing the recursive feasibility and closed-loop stability. Then, we propose the semi-explicit approach for move blocked MPC which combines the explicit approach for blocking position with simplified on-line optimization for blocked offset from the base sequence to overcome the above-mentioned second limitation. This allows the controller to apply the proper time-varying blocking structure according to the current state on-line. By this, we could efficiently improve the optimality of move blocked MPC with only a little additional computation cost for critical region search while guaranteeing the recursive feasibility and closed-loop stability.

Since model and data based approaches are complementary to each other, combination of MPC and machine learning (ML) is an emerging area of research. However, the existing studies only con-

sider the combination of nominal MPC with ML method, though researches for model-plant mismatch compensation have already been studied actively in offset-free MPC field to overcome the limitation of model based approach. Therefore, we propose model-plant mismatch learning offset-free model predictive control (MPC) which learns and exploits the intrinsic model-plant mismatch, and effectively combines the advantages of model and data based approaches and overcome the limitations of them. The model-plant mismatch is approximated by general regression neural network (GRNN) with supervised learning from the estimated steady-state disturbance for each set-point. An improved disturbance estimator is designed to exploit both the learned model-plant mismatch and stabilizing property of the nominal disturbance estimator. We also apply the learned model-plant mismatch to the target calculator and finite-horizon optimal control problem to improve the prediction accuracy and closed-loop performance of MPC. Moreover, we examine the robust asymptotic stability of the proposed offset-free MPC scheme, which is known to be really difficult in nominal offset-free MPC, by exploiting the learned model-plant mismatch information.

The summary of three proposed approaches are below:

- Move-blocked model predictive control with linear interpolation of base sequences.
- Move-blocked model predictive control with time-varying blocking structure by semi-explicit approach.
- Model-plant mismatch learning offset-free model predictive control.

The rest of this thesis is organised as follows. In Chapter 2, move blocked model predictive control with guaranteed stability and improved optimality using linear interpolation of base sequences is proposed. In Chapter 3, we propose move blocked model predictive control with improved optimality using semi-explicit approach for applying time-varying blocking structure. In Chapter 4, offset-free model predictive control with guaranteed robust asymptotic stability using model-plant mismatch learning is proposed. Finally, we present the conclusion of this thesis and summary in Chapter 5.

Chapter 2

Move-blocked model predictive control with linear interpolation of base sequences[1]

2.1 Introduction

One of the most outstanding issues in MPC is the online computational load required to solve the optimization problem within a sampling period. Computational load is not an issue for systems with relatively slow dynamics, intermittent input updates, or sufficient computational capacity. However, for systems with fast dynamics (e.g., vehicles, robots) or limited computational capacity (e.g., on-board single-chip controllers), computational load can hinder the real-time implementation of MPC. Therefore, various techniques have been proposed for reducing the online computational load. Some techniques are concerned with faster optimization algorithms by exploiting the structure of the optimal control problem. Among them are the hierarchical decomposition approach to separate the objective function of the optimization problem [9], warm-starting to obtain a better initial point of optimization algorithm from the information of the previous

[1] This chapter is a slightly adapted version of Son, S. H., Park, B. J., Oh, T. H., Kim, J. W., and Lee, J. M. (2019). "Move blocked model predictive control with guaranteed stability and improved optimality using linear interpolation of base sequences". International Journal of Control, under review.

sampling instant [10], and fast MPC to improve interior-point search direction by exploiting the structure of the optimization problem [11]. Other techniques reduce computational complexity by modifying the optimal control problem at the expense of optimality such as input parameterization. Various input parameterization techniques have been proposed including approximation of the control law with a parameterised function to reduce the order of problem [12, 13], a move blocking method fixes the decision variables over arbitrary time intervals [14], and linear interpolation of typical solutions [15]. Move blocking and linear interpolation are one of the most commonly used methods to reduce computational complexity, and are the focus of this study.

In input parameterization methods which construct input trajectory with a restricted number of variables, the analysis for closed-loop property such as feasibility and convergence is not straightforward as in [16] with only given degree of freedom (d.o.f). Therefore, these methods usually utilise the typical solutions with feasibility or stabilizing property, which can be derived without much computational cost, to improve the closed-loop performance of the controller.

In case of interpolation methods, they utilise these typical solutions directly as the basis. [17] use linear interpolation of unconstrainted linear quadratic regulator (LQR) solution and the shifted version of the previous solution as the solution of MPC. [18] interpolate LQR solution, tail of dual-mode control, and the mean level solution which is an easily computed feasible solution apart from the boundary of feasible solution set, to ensure feasibility under some uncertainty. [19] interpolate LQR solution, tail, and an explicit solution at the nearest facet of feasible region to reduce the compu-

tational complexity of multiparametric programming by restricting the required critical region. However, these interpolated solutions are usually too restrictive in general because the solutions have to lie in the convex hull of basis solutions. There also exist interpolation methods based on stabilizing feedback solutions. [20, 21, 22] directly construct the solution by interpolating feedback solutions, and [23] and [15] apply the interpolation at the terminal control law of dual-mode control. However, these methods use conservative constraint handling and the number of variables may become large for high dimensional systems.

In case of move blocking, they utilise the typical solutions as base sequece and block the offset from the base sequence. [14] use the infinite-horizon LQR solution as the base sequence to exploit the pre-stabilizing property and optimality. [24, 25] use the shifted previous solution as the base sequence to exploit the feasibility of the retained previous solution and guarantee monotonic decrease of the objective value. However, existing move blocking schemes use fixed base sequences only and do not fully exploit the valuable properties of several solutions. When using LQR solution as the base sequence, since its feasibility is not guaranteed, a specific format of blocking structure called moving window blocking proposed in [14] and a restrictive terminal constraint set must be used. On the other hand, when using the shifted previous solution as the base sequence, since the optimality of the solution is not guaranteed, the closed-loop performance of the controller can be degraded.

Thus, we propose to implement the interpolation of LQR solution and shifted previous solution as the base sequence of move blocking. Then, the solution trajectory of the proposed method can

be interpreted as the direct sum of interpolated solution and blocked offset term. This allows the controller to exploit the valuable properties of both parameterization methods and overcome each limitation by allowing both schemes to complement each other. In terms of interpolation method, the accessible space of the solution is expanded outside the convex hull of basis solutions by the blocked offset term with a flexible blocking structure. In terms of move blocking, the solution set embeds LQR solution and shifted previous solution by the base sequence, and the controller can utilise valuable properties of them. Therefore, the proposed method can effectively enlarge the feasible region, and improve the closed-loop optimality performance compared to the existing schemes while easily guaranteeing recursive feasibility, convergence, stability, only with a little additional computational cost due to the interpolation parameter.

The rest of this chapter is organised as follows. We provide the MPC formulation with move blocking for a discrete linear time-invariant system in *Section 2.2*. In *Section 2.3*, we analyze the existing move blocking schemes. In *Section 2.4*, we propose the interpolated solution based move bocking scheme, analyze the closed-loop properties of the proposed scheme, and provide the QP formulation of the interpolated solution based move blocked MPC. Finally, the numerical examples verify that the proposed interpolated solution based move blocking scheme can efficiently enlarge the feasible region and improve the optimality of move blocked MPC compare to the existing schemes in *Section 2.5*.

2.2 Preliminaries

2.2.1 MPC formulation

We consider the discrete linear time-invariant system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2.1}$$

where k is the discrete-time index, $u(k) \in \mathbb{R}^{n_u}$ is the input, $x(k) \in \mathbb{R}^{n_x}$ is the state, and $y(k) \in \mathbb{R}^{n_y}$ is the output. A, B, and C are matrices with appropriate dimensions, and the pair (A,B) is assumed to be controllable. n_u, n_x, and n_y are the dimensions of the input, state, and output vectors, respectively.

In the design of control systems, operational constraints are usually imposed for safe and stable operations. These constraints are commonly presented as convex polyhedral regions on the input and state variables.

$$u \in \mathcal{U},\ x \in \mathcal{X} \tag{2.2}$$

where $\mathcal{U} := \{u \in \mathbb{R}^{n_u} | A_u u \leq b_u\}$ and $\mathcal{X} := \{x \in \mathbb{R}^{n_x} | A_x x \leq b_x\}$ are compact polyhedral sets containing the origin in their interiors.

In addition, the terminal state $x_{_N}$ is usually imposed to lie in an control invariant terminal constraint set $\mathcal{X}_T$ to guarantee the recursive feasbility and closed-loop stability. A non-empty set $\mathcal{C} \in \mathcal{X}$ is referred to as a control invarinat (CI) set for the system in Eq. (2.1) subject to the constraints in Eq. (2.2), if and only if $\forall x \in \mathcal{C},\ \exists u \in \mathcal{U}$ such that $Ax + Bu \in \mathcal{C}$ [26].

The objective function of regulation problem is commonly described as a sum of quadratic stage costs:

$$J(x_0, U) := x_N^\top Q_N x_N + \sum_{i=0}^{N-1} \phi(x_i, u_i) \tag{2.3}$$
$$\phi(x, u) := x^\top Q_x x + u^\top Q_u u$$

where $\phi(x, u)$ denotes a single-stage cost. $Q_x \in \mathbb{R}^{n_x \times n_x}$, $Q_N \in \mathbb{R}^{n_x \times n_x}$, and $Q_u \in \mathbb{R}^{n_u \times n_u}$ are positive definite weighting matrices for the state vector, terminal state vector, and input vector, respectively.

Then, the optimal control problem in **P1** is solved at each sampling instant to derive an optimal input sequence.

$$\begin{aligned} \mathbf{P1}: \ J^*(x) := \min_U \quad & J(x_0, U) \\ \text{s.t.} \quad & x_0 = x, \ x_{i+1} = Ax_i + Bu_i \\ & u_i \in \mathcal{U}, \ x_{i+1} \in \mathcal{X}, \ \forall i = 0, \cdots, N-1 \\ & x_N \in \mathcal{X}_T \end{aligned}$$

where U denotes the future input sequence $[u_{0|k}; u_{1|k}; \ \cdots \ ; u_{N-1|k}]$ ($[v_1; v_2; \cdots ; v_n]$ denotes the vertical concatenation $[v_1^\top, v_2^\top, \cdots, v_n^\top]^\top$).

The objective function in Eq. (2.3) can be rewritten as a quadratic function of U by expressing all the predicted states explicitly:

$$J(x, U) = U^\top H U + 2U^\top f + c \tag{2.4}$$

$$H := \Psi^\top Q_{_X} \Psi + Q_{_U}, \quad f := \Psi^\top Q_{_X} \Phi x$$
$$c := x^\top (\Phi^\top Q_{_X} \Phi + Q_x) x$$
$$\Phi := \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Psi := \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}$$

where $Q_{_X} \in \mathbb{R}^{Nn_x \times Nn_x}$ and $Q_{_U} \in \mathbb{R}^{Nn_u \times Nn_u}$ are the matrices with diagonal form having $\{Q_x, \cdots, Q_x, Q_{_N}\}$ and $\{Q_u, \cdots, Q_u\}$ as main diagonal blocks, respectively. Since $\Psi^T Q_{_X} \Psi \succ 0$, $Q_{_U} \succ 0$, and $H \succ 0$, the objective function J in Eq. (2.4) is a positive definite quadratic function of U.

Since $\mathcal{U}$, $\mathcal{X}$, and $\mathcal{X}_T$ are polytopic, we can describe constraints in **P1** as a set of linear inequalities with suitable matrices, and then **P1** can be reformulated in a condensed form with the objective function in Eq. (2.4):

$$\begin{aligned} \mathbf{P1}' : \ J^*(x) := \min_{U} \quad & J(x, U) \\ \text{s.t.} \quad & Fx + GU \leq h \end{aligned}$$

where $F \in \mathbb{R}^{n_c \times n_x}$, $G \in \mathbb{R}^{n_c \times Nn_u}$ and $h \in \mathbb{R}^{n_c}$ are suitable matrices, and n_c is the number of inequalities.

In MPC, $\mathbf{P1}'$ is solved based on the current measured(or observed) state for each sampling instant, then the first input of the resulting optimal input trajectory is implemented on the system. This procedure is repeated at later sampling instants in a receding horizon

fashion.

2.2.2 Move blocking

Since the computational complexity of solving an optimization problem depends on the number of decision variables or d.o.f., the move blocking method that reduces the number of decision variables by fixing the decision variables over blocks illustrated in **Figure 2.1** is commonly used in practice. The input blocking directly parameterizes the input sequence, while the offset blocking parameterizes the offset from the base sequence.

The structure of move blocking is determined by the blocking positions in **Definition 1**.

Definition 1. (Blocking position set) The set $s := \{s_1, \cdots, s_{\overline{N}}\}$ in ascending order denotes the blocking position set, where each component $s_i \in \mathbb{N}_{\leq N}$ denotes the blocking position where each block begins, and $\overline{N}$ is the number of blocks ($\mathbb{N}_{\leq N}$ is the set of natural numbers less than or equal to N). Additionally, we represent the set of admissible s as $\mathcal{S}$.

Definition 2. (Blocking matrix) $\overline{P} \in \mathbb{B}^{N \times \overline{N}}$ denotes the blocking matrix, where $\mathbb{B}^{N \times \overline{N}}$ is an $N \times \overline{N}$ matrix whose elements are restricted to the binary values 0 or 1. $\overline{P}$ is a lower triangular matrix, and the position of non-zero elements in $\overline{P}$ is determined by the blocking position set s:

$$\overline{P} = [\overline{P}_1, \cdots, \overline{P}_{\overline{N}}] \tag{2.5}$$

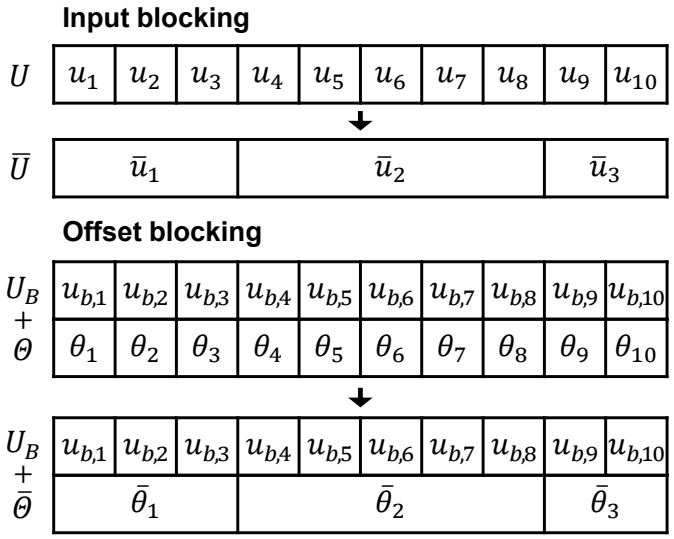

Figure 2.1: A schematic illustration of move blocking when $N = 10$, $\overline{N} = 3$, and $s = \{1, 4, 9\}$.

$$\overline{P}_i := \begin{cases} [\mathbf{0}_{s_i-1}; \mathbf{1}_{s_{i+1}-s_i}; \mathbf{0}_{N-s_{i+1}+1}] & \text{for} \quad i = 1, \cdots, \overline{N}-1 \\ [\mathbf{0}_{s_i-1}; \mathbf{1}_{N-s_i+1}] & \text{for} \quad i = \overline{N} \end{cases}$$

where $\overline{P}_i$ is the i^{th} column of $\overline{P}$, s_i is the i^{th} component of s, and $\mathbf{0}_m$ and $\mathbf{1}_n$ represent a vector of zeros with length m and a vector of ones with length n, respectively.

In case of input blocking, the entire input sequence U can be described with $\overline{P}$ and a reduced input sequence $\overline{U} := [\overline{u}_1; \overline{u}_2; \cdots; \overline{u}_{\overline{N}}]$ as

$$U = (\overline{P} \otimes I_{n_u})\overline{U} = P\overline{U} \tag{2.6}$$

where $P := \overline{P} \otimes I_{n_u}$ and $\otimes$ denotes the Kronecker product.

In case of offset blocking, U is formulated with a base sequence $U_B \in \mathbb{R}^{Nn_u}$ and an offset sequence $\Theta := [\theta_1; \theta_2; \cdots; \theta_N]$. Then, Θ can be parameterized with move blocking as

$$U = U_B + \Theta = U_B + P\overline{\Theta} \tag{2.7}$$

where $\overline{\Theta}$ denotes a reduced offset sequence $[\overline{\theta}_1; \overline{\theta}_2; \cdots; \overline{\theta}_{\overline{N}}]$.

Despite the studies aimed at improving the performances of input blocking [16, 27], input blocking suffers from poor closed-loop performance owing to the inflexibility associated with fixing the actual inputs [14]. On the other hand, offset blocking fixes the offset from the base sequence not the actual input.

2.2.3 Move blocked MPC (MBMPC)

In this study, we proceed the discussion based on the MPC with the offset blocking scheme in Eq. (2.7). When move blocking is implemented on MPC, the d.o.f. is reduced by parameterization, while the blocking positions are added as new variables with the additional constraints as in **P2**.

$$
\begin{aligned}
\mathbf{P2}: \ J^*(x, U_B) := \min_{s,\overline{\Theta}} \quad & J(x_0, U) \\
\text{s.t.} \quad & U = U_B + P\overline{\Theta} \\
& x_0 = x, \ x_{i+1} = Ax_i + Bu_i \\
& u_i \in \mathcal{U}, \ x_{i+1} \in \mathcal{X}, \ x_{_N} \in \mathcal{X}_T \\
& s = \{s_1, \cdots, s_{\overline{N}}\}, \ s_j \in \mathbb{N}_{\leq N} \\
& \forall i = 0, \cdots, N-1, \ \forall j = 1, \cdots, \overline{N}
\end{aligned}
$$

The objective function in Eq. (2.4) can be reformulated as a quadratic function for the parameterized offset $\overline{\Theta}$:

$$
\begin{aligned}
J(x, U_B, s, \overline{\Theta}) &= \overline{\Theta}^\top \overline{H\Theta} + 2\overline{\Theta}^\top \overline{f} + \overline{c} \qquad (2.8) \\
\overline{H} &:= P^\top H P \\
\overline{f} &:= P^\top (HU_B + f) \\
\overline{c} &:= U_B^\top H U_B + 2U_B^\top f + c
\end{aligned}
$$

Then, **P2** can be reformulated in a compact form using Eq. (2.8):

$$\begin{aligned}
\mathbf{P2}' : \; & J^*(x,U_B) = \min_{s,\overline{\Theta}} \; J(x, U_B, s, \overline{\Theta}) \\
\text{s.t.} \quad & \overline{F}x + \overline{G}\overline{\Theta} \leq \overline{h} \\
& s = \{s_1, \cdots, s_{\overline{N}}\} \\
& s_i \in \mathbb{N}_{\leq N}, \; \forall i = 1, \cdots, \overline{N}
\end{aligned}$$

where $\overline{F} \in \mathbb{R}^{\overline{n}_c \times n_x}$, $\overline{G} \in \mathbb{R}^{\overline{n}_c \times N n_u}$ and $\overline{h} \in \mathbb{R}^{\overline{n}_c}$ are appropriate matrices, and $\overline{n}_c$ is the number of inequalities.

2.3 Move blocking schemes

In nominal MPC, the CI terminal constraint set is imposed to ensure recursive feasibility and stability as in *Section 2.2.1*. However, constraining the $x_{_N}$ to lie in a CI set cannot guarantee recursive feasibility in MBMPC owing to the extra constraints imposed on the control input sequence by move blocking. Therefore, several move blocking schemes have been developed to ensure recursive feasibility.

One input blocking scheme directly allows the closed-loop state trajectory to remain within a feasible set by proposing the notion of a control invariant feasibility (CIF) set [27]. However, this input blocking scheme has several major drawbacks such as flexibility degradation of the controller owing to the blocking of the actual input and considerable domain reduction owing to the restrictiveness of the CIF set.

In contrast to input blocking, offset blocking guarantee recursive feasibility and stability by utilizing the valuable property from

the base sequence. We introduce two representative offset blocking schemes in subsequent sections.

2.3.1 Previous solution based offset blocking

[24, 25] formulate an offset blocking scheme which retains the previous solution for the base sequence at the current sampling instant, and parameterizes the input sequence in terms of the offsets from it.

$$U = \lambda \widehat{U}(k) + P\overline{\Theta} \tag{2.9}$$
$$\widehat{U}(k) := [u^*_{1|k-1}; \cdots ; u^*_{N-1|k-1}; \mathbf{0}_{n_u}]$$

where $\lambda \in [0, 1]$, and $[u^*_{1|k-1}; \cdots ; u^*_{N-1|k-1}]$ represents the shifted input solution from the previous sampling instant.

Since the shifted version of previous solution sequence is always constructible from the base sequence, this scheme can easily ensure the recursive feasibility and stability with the terminal constraint set of nominal MPC, i.e., the maximal control invariant set in **Definition 3**, for all admissible blocking structure only with the condition $s_{\overline{N}} = N$. Therefore, this scheme can handle a relatively large feasible region compared to other move blocking schemes where the previous solution is not constructible.

Definition 3. (Maximal control invariant set) The maximal control invariant (MCI) set $\mathcal{C}_\infty$ is a CI set which contains all other CI sets

and can also be defined as

$$\mathcal{C}_\infty := \{x_0 \in \mathcal{X} \mid \exists \{u_i \in \mathbb{R}^{n_u}\}_{i=0}^{\infty} \text{ such that } u_i \in \mathcal{U}, \\ x_{i+1} = Ax_i + Bu_i \in \mathcal{X} \ \ \forall i \in \mathbb{Z}_{\geq 0}\} \tag{2.10}$$

where $\mathbb{Z}_{\geq 0}$ is the set of non-negative integers.

However, since the optimality of the retained previous solution is not ensured, the base sequence can be undesirable for the current time step especially when the solution sequence is not sufficiently updated by the blocked offset term at the early stage of the control system or the reference trajectory rapidly changes in a tracking problem. In this case, the optimality performance of the controller can be degraded. Moreover, this scheme cannot utilise the stabilizing property of proper feedback law such as the LQR solution.

2.3.2 LQR solution based offset blocking

[14] proposed an offset blocking scheme which utilises the infinite-horizon LQR solution as the base sequence. The entire input sequence can be formulated as

$$U = U_{LQR}(k) + \Psi_{_L} P\overline{\Theta} \tag{2.11}$$

$$
\begin{aligned}
&U_{LQR}(k) := \Phi_{_L} x(k) \\
&\Psi_{_L} := \begin{bmatrix} I_{n_u} & 0 & 0 & \cdots & 0 \\ K_{_L}B & I_{n_u} & 0 & \cdots & 0 \\ K_{_L}A_{_L}B & K_{_L}B & I_{n_u} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ K_{_L}A_{_L}^{N-2}B & K_{_L}A_{_L}^{N-3}B & K_{_L}A_{_L}^{N-4}B & \cdots & I_{n_u} \end{bmatrix} \\
&\Phi_{_L} := \begin{bmatrix} K_{_L} \\ K_{_L}A_{_L} \\ \vdots \\ K_{_L}A_{_L}^{N-1} \end{bmatrix}, \; A_{_L} := A + BK_{_L}
\end{aligned}
$$

where $K_{_L}$ is the infinite-horizon LQR gain. $U_{LQR}(k)$ is the LQR solution sequence from $x(k)$.

The underlying philosophy of this scheme is quite similar to the dual-mode control in [28, 29]. This scheme can exploit the optimality and stabilizing property of the LQR solution. Therefore, when the state enters the maximal positive invariant (MPI) set subject to the LQR gain in **Definition 4**, this scheme can stabilise the system in an optimal way.

Definition 4. (Maximal positive invariant set) The set $O_\infty(K_s)$ is the maximal positive invariant (MPI) set for the system in Eq. (2.1) subject to a stabilizing feedback gain K_s [30].

$$
\begin{aligned}
\mathcal{O}_\infty(K_s) := \{x_0 \in \mathcal{X} \mid x_{i+1} = (A + BK_s)x_i \in \mathcal{X}, \\
u_i = K_s x_i \in \mathcal{U} \;\; \forall i \in \mathbb{Z}_{\geq 0}\}
\end{aligned} \tag{2.12}
$$

However, when the state is outside MPI set subject to $K_{_L}$ and thus the LQR solution is not feasible, the shifted version of previous solution sequence has to be constructible at the current step to sufficiently ensure the recursive feasibility. For this, the blocking structure of this scheme is restricted to propagate through a specific pattern called moving window blocking (MWB) strategy and the terminal state has to lie in the relatively restrictive set, $O_\infty(K_{_L})$, compared to that of nominal MPC, $\mathcal{C}_\infty$. Therefore, this scheme can only handle the smaller feasible region than other move blocking schemes with flexibility in selecting the blocking structure and a larger terminal set.

2.4 Interpolated solution based move blocking

The offset blocking schemes introduced in *Section 2.4* mainly focus on utilizing the fixed form of base sequences, the previous solution $\widehat{U}$ and the LQR solution U_{LQR}, respectively. Since $\widehat{U}$ and U_{LQR} have their own advantages and limitations as base sequences, we propose to implement the interpolated solution of $\widehat{U}$ and U_{LQR} as the base sequence to improve the closed-loop performance of the MBMPC by addressing the limitations and inheriting the valuable properties of each solution.

2.4.1 Interpolated solution based MBMPC

The proposed interpolated solution based offset blocking uses the linear interpolation of $\widehat{U}$ and U_{LQR} as the base sequence.

$$U_B(k) = \lambda \widehat{U}(k) + (1-\lambda) U_{LQR}(k) \tag{2.13}$$

where $\lambda \in [0, 1]$ is the interpolation parameter.

The input sequence within the optimization window is the sum of the base sequence and the parameterized offset expressed as

$$U = \lambda \widehat{U}(k) + (1 - \lambda) U_{LQR}(k) + P\overline{\Theta} \tag{2.14}$$

The optimal control problem of interpolated solution based MBMPC is given by

$$\begin{aligned}
\mathbf{P3}: \ J^*(x, \widehat{U}) := \min_{s,\overline{\Theta},\lambda} \quad & J(x_0, U) \\
\text{s.t.} \quad & U = \lambda \widehat{U}(k) + (1 - \lambda) U_{LQR}(k) + P\overline{\Theta} \\
& x_0 = x, \ x_{i+1} = Ax_i + Bu_i \\
& u_i \in \mathcal{U}, \ x_{i+1} \in \mathcal{X}, \ x_{_N} \in \mathcal{X}_T \\
& s = \{s_1, \cdots, s_{\overline{N}}\}, \ s_j \in \mathbb{N}_{\leq N} \\
& \forall i = 0, \cdots, N-1, \ \forall j = 1, \cdots, \overline{N} \\
& 0 \leq \lambda \leq 1
\end{aligned}$$

Since the shifted optimal solution sequence at the previous sampling instant $[u^*_{1|k-1}; \cdots; u^*_{N-1|k-1}]$ can be constructed from $\widehat{U}$ in Eq. (2.14), this scheme guarantees the recursive feasibility and stability. These closed-loop properties can be proved in a similar way to nominal MPC in [1].

Theorem 1. Consider the optimal control problem **P3**. If the blocking position set s satisfies $s_{\overline{N}} = N$, then a feasible solution is always guaranteed to exist.

Proof) We can select the interpolation parameter as $\lambda = 1$ and the reduced offset sequence as $\overline{\Theta} = [0; \cdots; 0; \nu]$, $\nu \in \mathcal{U}$. If $s_{\overline{N}} = N$, then the entire offset sequence can have the form of $\Theta = [0; \cdots; 0; \nu]$, and the entire input sequence can always be given by

$$\widetilde{U}(k) := [u^*_{1|k-1}; \cdots; u^*_{N-1|k-1}; \nu] \tag{2.15}$$

which results in the prediction of future state sequence as

$$\widetilde{X}(k) := [x^*_{2|k-1}; \cdots; x^*_{N|k-1}; \widetilde{x}_{N|k}] \tag{2.16}$$

Since $[u^*_{1|k-1}; \cdots; u^*_{N-1|k-1}]$ is the shifted version of the previous optimal input solution, $x^*_{N|k-1} \in C_\infty$ is guaranteed. Moreover, because of the invariance of C_∞, there always exists $\nu \in \mathcal{U}$ such that $\widetilde{x}_{N|k} = Ax^*_{N|k-1} + B\nu \in C_\infty$. Therefore, we can ensure that there always exists a feasible solution for the problem. □

Remark 1. From **Theorem 1**, we can reformulate any arbitrary input sequence U in Eq. (2.14) as in Eq. (2.17) with a pseudo base sequence $\widetilde{U}_B$.

$$U = \widetilde{U}_B(k) + P\overline{\Theta}_\nu \tag{2.17}$$

$$\begin{aligned}
\widetilde{U}_B(k) &:= \lambda\widetilde{U}(k) + (1-\lambda)U_{LQR}(k) \\
\overline{\Theta}_\nu &:= [\overline{\theta}_1; \overline{\theta}_2; \cdots; \overline{\theta}_{\overline{N}} - \lambda\nu]
\end{aligned}$$

From **Remark 1**, we can consider $\widetilde{U}_B$ as the base sequence of the interpolated solution based MBMPC instead of U_B, and proceed with the analysis using $\widetilde{U}_B$.

Remark 2. Since $\widetilde{U}$ in Eq. (2.15) is ensured to be feasible with proper $\nu \in \mathcal{U}$, the base sequence $\widetilde{U}_B$ always enters the feasible solution set as $\lambda \to 1$ even when U_{LQR} and $\widehat{U}$ are infeasible.

In case of LQR solution based MBMPC, since the LQR solution is not guaranteed to be feasible, a specific pattern of blocking structure called MWB strategy and a restrictive terminal set $\mathcal{O}_\infty(K_{_L})$ have to be used to ensure recursive feasibility and stability, which can considerably reduce the feasible region. On the other hand, in case of the proposed interpolated solution based MBMPC, since the feasibility of **P3** is ensured by the base sequence as in **Remark 2**, the selection of blocking structure is flexible only with the condition of $s_{\overline{N}} = N$. Moreover, since LQR solution does not need to be feasible at the terminal state, the proposed scheme does not require to impose the terminal state to lie in $\mathcal{O}_\infty(K_{_L})$ and can use a relatively large terminal constraint set, $\mathcal{C}_\infty$. Therefore, the proposed scheme can handle a relatively large feasible region compared to that of LQR solution based MBMPC. We demonstrate this property by directly comparing the feasible region of both schemes in the numerical example section.

In case of previous solution based MBMPC, since the optimality of the base sequence is not guaranteed, the optimality performance of the controller can be considerably degraded than that of the nominal MPC. On the other hand, in case of the proposed interpolated solution based offset blocked MPC, since the base sequence $\widetilde{U}_B$ is always accessible to the infinite-horizon LQR solution as $\lambda \to 0$, the controller has an additional d.o.f. in the monotonically decreasing direction of the objective value. Therefore, this scheme can effectively improve

the optimality performance of the controller by adjusting the interpolation parameter with the parameterised offset compared to the simple previous solution based MBMPC. Moreover, we can maximise the utilization of the closed-loop optimality of the LQR solution in this scheme by applying the concept of dual-mode prediction in [29] and fixing the control input as the LQR solution when the state of the system reaches the $\mathcal{O}_\infty(K_{_L})$ where the LQR solution is always feasible and optimal.

The convergence property of the interpolated solution based MBMPC is described below.

Theorem 2. Consider the closed-loop receding horizon control system with input update by the optimal control problem of the interpolated solution based MBMPC in **P3**. The system will converge to the origin as $k \to \infty$ under the commonly used basic stability assumption given by

$$\min_{\nu \in \mathcal{U}, \widetilde{x}_{N|k} \in \mathcal{C}_\infty} {x^*_{N|k-1}}^T (Q_x - Q_{_N}) x^*_{N|k-1} + \phi_N(\widetilde{x}_{N|k}, \nu) \leq 0 \quad (2.18)$$

where $\phi_N(x, u) := x^T Q_{_N} x + u^T Q_u u$.

Proof) As in the proof of **Theorem 1**, we can obtain a feasible solution $\widetilde{U}$ in Eq. (2.15) by selecting s with $s_{\overline{N}} = N$, $\lambda = 1$, and $\overline{\Theta} = [0; \cdots; 0; \nu]$. Then, we can describe the expected objective value

when $\widetilde{U}$ is implemented to the system as

$$J(x(k), \widetilde{U}(k)) = \sum_{i=1}^{N-1} \phi(x^*_{i|k-1}, u^*_{i|k-1}) + \phi(x^*_{N|k-1}, \nu) + \widetilde{x}^\top_{N|k} Q_{_N} \widetilde{x}_{N|k} \tag{2.19}$$

where $x^*_{1|k-1} = x(k)$.

Since the optimal solution sequence of the previous sampling instant $U^*_{k-1} := [u^*_{0|k-1}; u^*_{1|k-1}; \cdots ; u^*_{N-1|k-1}]$ and $\widetilde{U}$ are identical except for the first and last components, we can reformulate Eq. (2.19) as

$$\begin{aligned} J(x(k), \widetilde{U}(k)) =& J^*_{k-1} - \phi(x(k), u(k-1)) \\ &+ {x^*_{N|k-1}}^{T} (Q_x - Q_{_N}) x^*_{N|k-1} + \phi_N(\widetilde{x}_{N|k}, \nu) \end{aligned} \tag{2.20}$$

where $J^*_i := J^*(x(i), \widehat{U}(i), s(i))$ denotes the optimal cost at the sampling instant i.

Since $\widetilde{U}$ is not the optimal input sequence, the inequality in Eq. (2.21) holds:

$$J^*_k \leq J(x(k), \widetilde{U}(k)) \tag{2.21}$$

Substituting Eq. (2.20) into Eq. (2.21) gives

$$\begin{aligned} J^*_k \leq J^*_{k-1} &- \phi(x(k), u(k-1)) \\ &+ {x^*_{N|k-1}}^{T} (Q_x - Q_{_N}) x^*_{N|k-1} + \phi_N(\widetilde{x}_{N|k}, \nu) \end{aligned} \tag{2.22}$$

By the assumption in Eq. (2.18), Eq. (2.22) can be reformulated as

$$J_k^* - J_{k-1}^* \leq -\phi(x(k), u(k-1)) \tag{2.23}$$

Since $\phi(x(k), u(k-1)) > 0 \; \forall x(k) \neq \mathbf{0}_{n_x}$ and $u(k-1) \neq \mathbf{0}_{n_u}$, we can see that the optimal cost J^* strictly decreases over time. Moreover, since the optimal cost is lower-bounded by zero, $J^* \geq 0$, the sequence of J^* converges to zero as the state of the closed-loop system converges to the origin. □

The closed-loop stability of the interpolated solution based MBMPC is easily ensured by **Corollary 1**.

Corollary 1. When the convergence to the origin of the interpolated solution based MBMPC is guaranteed, the asymptotic stability of the origin is inherently ensured by the fact that $\mathcal{O}_\infty(K_{_L})$ is a neighborhood containing the origin in its interior, and the stabilizing control law $u = K_{_L} x$ is feasible and optimal $\forall x \in \mathcal{O}_\infty(K_{_L})$.

2.4.2 QP formulation

We reformulate the input sequence in Eq. (2.14) as

$$U = U_{LQR}(k) + M(k)V \tag{2.24}$$

$$M(k) := [\widehat{U}(k) - U_{LQR}(k),\ P]$$
$$V := [\lambda; \overline{\Theta}]$$

where V is the vector of decision variables.

Then, the objective function can be reformulated in a quadratic function for V:

$$J = V^T \overline{H}_{_V} V + 2V^T \overline{f}_{_V} + \overline{c}_{_V} \tag{2.25}$$

$$\begin{aligned}
\overline{H}_{_V} &:= M(k)^T H M(k) \\
\overline{f}_{_V} &:= M(k)^T (H U_{LQR}(k) + f) \\
\overline{c}_{_V} &:= U_{LQR}(k)^T H U_{LQR}(k) + 2U_{LQR}(k)^T f + c
\end{aligned}$$

The constraints can be formulated in terms of V as Eqs. (2.26)–(2.30). For generality, the constraint on the rate of input changes $\delta u \in \mathcal{U}_d$ is also considered, where $\mathcal{U}_d := \{\delta u \in \mathbb{R}^{n_u} : A_{\delta u} \delta u \leq b_{\delta u}\}$ is a compact polyhedral set containing the origin in the interior.

- Input values

$$[(I_{_N} \otimes A_u) M(k)] V \leq C_1 \otimes b_u - (I_{_N} \otimes A_u) U_{LQR}(k) \tag{2.26}$$

- Rate of input changes

$$\begin{aligned}
[(I_{_N} \otimes A_{\delta u}) C_2 M(k)] V &\leq C_1 \otimes b_{\delta u} \\
&\quad - (I_{_N} \otimes A_{\delta u})(C_2 U_{LQR}(k) - \mathbf{u}_0(k))
\end{aligned} \tag{2.27}$$

- State values

$$\begin{aligned}
[(I_{_N} \otimes A_x) \Psi M(k)] V &\leq C_1 \otimes b_x \\
&\quad - (I_{_N} \otimes A_x)(\Phi x(k) + \Psi U_{LQR}(k))
\end{aligned} \tag{2.28}$$

- Terminal state value

$$[A_T C_3 \Psi M(k)] V \leq b_T - A_T C_3 (\Phi x(k) + \Psi U_{LQR}(k))$$

▪ Weighting factor

$$\begin{bmatrix} -C_4 \\ C_4 \end{bmatrix} V \leq \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.29}$$

where

$$\mathbf{u}_0(k) := \begin{bmatrix} u(k-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \; C_1 := \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \; C_2 := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \otimes I_{n_u}$$

$$C_3 := [\mathbf{0}_{n_x \times n_x}, \cdots, \mathbf{0}_{n_x \times n_x}, I_{n_x}], \; C_4 := [1, 0, \cdots, 0]$$

where $\mathbf{0}_{n_x \times n_x}$ represents the $n_x \times n_x$ matrix of zeros.

Now, after the blocking structure is selected, we can reformulate **P3** as a standard QP problem by substituting Eqs. (2.25)–(2.30) into **P3** and omitting the constant term:

$$\begin{aligned} \mathbf{P4}: \; J^*(x(k), \widehat{U}(k), s) = \min_V \quad & V^T \overline{H} V + 2 V^T \overline{f} \\ \text{s.t.} \quad & \Gamma V \leq \gamma \end{aligned}$$

where

$$\Gamma := \begin{bmatrix} (I_N \otimes A_u)M(k) \\ (I_N \otimes A_{\delta u})C_2 M(k) \\ (I_N \otimes A_x)\Psi M(k) \\ A_T C_3 \Psi M(k) \\ -C_4 \\ C_4 \end{bmatrix}$$

$$\gamma := \begin{bmatrix} C_1 \otimes b_u - (I_N \otimes A_u)U_{LQR}(k) \\ C_1 \otimes b_{\delta u} - (I_N \otimes A_{\delta u})(C_2 U_{LQR}(k) - \mathbf{u}_0(k)) \\ C_1 \otimes b_x - (I_N \otimes A_x)(\Phi x(k) + \Psi U_{LQR}(k)) \\ b_T - A_T C_3(\Phi x(k) + \Psi U_{LQR}(k)) \\ 0 \\ 1 \end{bmatrix}$$

Then, the optimal control problem in **P4** can be solved by a QP solver. The outline of the interpolated solution based MBMPC with applying the concept of dual-mode control is summarised in **Algorithm 2.1**.

2.5 Numerical examples

In this section, we demonstrate the efficacy of the interpolated solution based MBMPC through examples comparing the performance of the proposed scheme and existing MBMPC schemes in terms of the volume of feasible region, closed-loop cost and computational load.

The selection of blocking positions has a significant impact on

Algorithm 2.1. Interpolated solution based MBMPC

Initialise $x(0) = x_0$, $\widehat{U}(0) = \mathbf{0}_{Nn_u}$
for $k = 0, 1, \cdots, K$ **do**
  Measure(or estimate) $x(k)$ at sampling instant k
  Select blocking position set $s(k)$ with $s_{\overline{N}} = N$
  Compute $U_{LQR}(k) = \Phi_{_L} x(k)$
  if $x(k) \in \mathcal{O}_\infty(K_{_L})$ **then**
    Set $\lambda^* = 0$, $\overline{\Theta}^* = \mathbf{0}_{\overline{N}n_u}$
  else
    Solve **P4** $\rightarrow \lambda^*$, $\overline{\Theta}^*$
  end if
  Compute $U^* = \lambda^* \widehat{U}(k) + (1 - \lambda^*) U_{LQR}(k) + P\overline{\Theta}^*$
  Apply $u^*_{0|k}$ to the system
  Update $\widehat{U}(k+1) = [u^*_{1|k}; \cdots; u^*_{N-1|k}; \mathbf{0}_{n_u}]$
  Wait for the next sampling instant $k+1$
end for

Table 2.1: Interpolated solution based MBMPC

the optimality performance of the controller, and the optimality of the blocking structure has been considered in various perspectives [25, 31]. However, in this study, since we only focus on the efficiency of base sequence, we apply an arbitrary time-invariant blocking structure for all the cases.

The terminal MPI set $O_\infty(K_{_L})$ and MCI set $\mathcal{C}_\infty$ were calculated using the MPT3 toolbox [32]. Simulations are performed using MATLAB® R2019a with Intel® Core™ i7-6700 CPU @ 3.40GHz, 32 GB RAM.

2.5.1 Example 1 (Feasible region)

The linear discrete time model is given by

$$x(k+1) = \begin{bmatrix} 0.9146 & 0.1665 & 0.0405 \\ 0.2665 & 0.3353 & 0.0058 \\ 0 & 0.0405 & 0.5353 \end{bmatrix} x(k) + \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0 \end{bmatrix} u(k)$$

The system constraints are given by

$$-100 \leq x_1 \leq 100,\ -100 \leq x_2 \leq 100,\ -100 \leq x_3 \leq 100$$
$$-1 \leq u_1 \leq 1,\ -2 \leq u_2 \leq 2$$

We derived the feasible region of each MBMPC scheme with $N = 10$, $\overline{N} = 3$, and the time-invariant blocking structure $s = \{1, 5, 10\}$. The control parameters are $Q_x = diag\{2, 2, 2\}$ and $Q_u = diag\{1, 1\}$. We also describe the trajectory of a constrained regulation problem from the initial state $x(0) = [60; -60; 60]$ with the proposed scheme.

Figure 2.2 shows the feasible regions of the nominal MPC and MBMPC based on previous solution, interpolated solution, and LQR solution. The first three 2D plots are the projections of the 3D feasible region in the last plot onto 2D spaces, respectively. It is clear that the feasible region of the proposed interpolated solution based MBMPC is much larger than that of LQR solution based MBMPC. The considerable difference in the volume of the feasible regions is mainly due to the difference in the terminal sets. The LQR based MBMPC have to constrain the terminal state in the MPI set for recursive feasibility. On the other hand, since previous solution or interpolated solution based MBMPC can inherently ensure the recursive feasibility by the

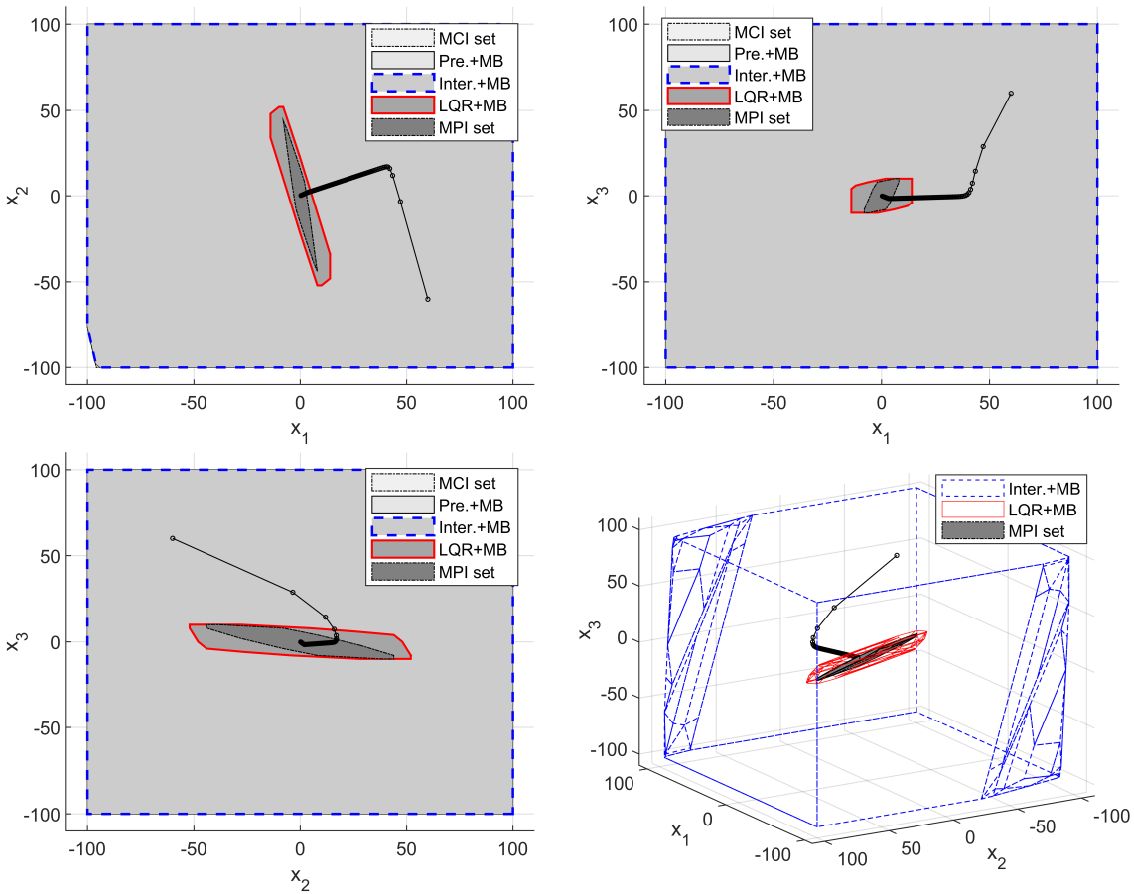

Figure 2.2: Feasible regions of the nominal MPC and MBMPC based on previous solution, interpolated solution, and LQR solution.

base sequence, it has the terminal MCI set yielding a much larger feasible region.

2.5.2 Example 2 (Performance in regulation problem)

The linear descrete time model of the ball-plate system in [14] is given by

$$x(k+1)=\begin{bmatrix} 1 & 0.03 & -0.315 & -0.00412 \\ 0 & 1 & -21 & -0.452 \\ 0 & 0 & 1 & 0.0514 \\ 0 & 0 & 0 & 2.71 \end{bmatrix} x(k) + \begin{bmatrix} -0.00011 \\ -0.0156 \\ 0.00245 \\ 0.195 \end{bmatrix} u(k)$$

$x = [l_b, v_b, \phi, v_\phi]$ denotes the state vector where l_b and v_b are the position and velocity of the ball, and ϕ and v_ϕ are the angle and angular velocity of the plate, respectively. The input variable u is the voltage to the motor of the plate. The operational constraints are given by

$$-20 \leq l_b \leq 20,\ -30 \leq v_b \leq 30,\ -10 \leq \phi \leq 10$$
$$-2 \leq v_\phi \leq 2,\ -10 \leq u \leq 10$$

We simulated a constrained regulation problem from the initial state $x(0) = [15; 5; -0.1; 1]$ using nominal MPC and the proposed and existing MBMPC scheme with $N = 15$, $\overline{N} = 3$, and the time-invariant blocking structure $s = \{1, 8, 15\}$. The control parameters are $Q_x = diag\{6, 0.1, 500, 100\}$, $Q_N = 3Q_x$, and $Q_u = 1$.

Figure 2.3 shows the results of nominal MPC and MBMPC based on the interpolated solution and previous solution in regulation problem. In this example, the states and input trajectories of the proposed

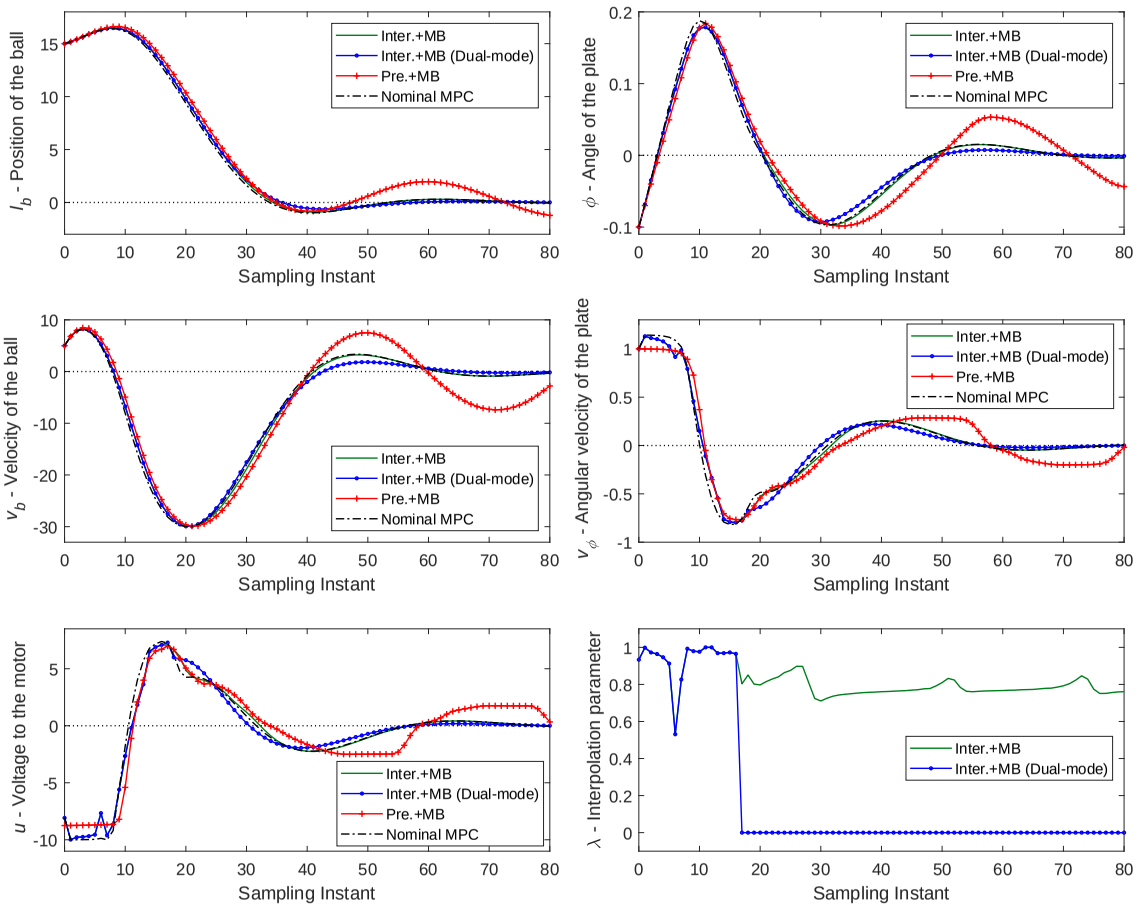

Figure 2.3: Comparison of the closed-loop performance of nominal MPC and MBMPC based on the interpolated solution with and without dual-mode control, and previous solution in regulation problem in the ball-plate system.

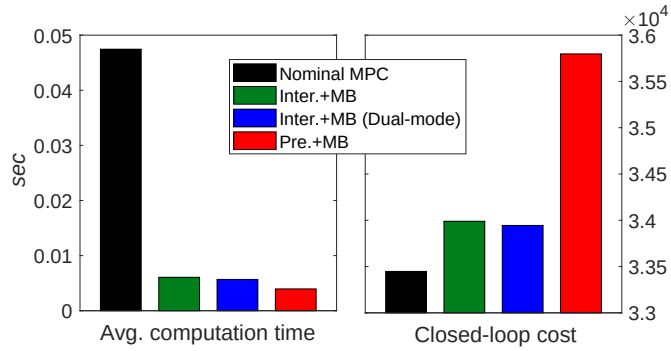

Figure 2.4: Comparison of the average computation time and closed-loop cost of nominal MPC and MBMPC based on the interpolated solution with and without dual-mode control, and previous solution in regulation problem in the ball-plate system.

interpolated solution based MBMPC are regulated to the origin faster than that of the existing previous solution based MBMPC. Moreover, we also compare the result of interpolated solution based MBMPCs with and without the dual-mode control. We can see the interpolation parameter λ of the case with dual-mode control is maintained as 0 from the 17^{th} sampling instant, after the state of the system reaches $\mathcal{O}_{\infty}(K_{_L})$, to fully utilise the optimality of LQR solution. Since the infinite horizon LQR solution is superior to that of the finite horizon optimal solution, if it is feasible, the interpolated solution based MBMPC with dual-mode control even shows better performance than that of the nominal MPC after the 17^{th} sampling instant. On the other hand, the interpolated solution based MBMPC without dual-mode control cannot fully exploit the optimality of LQR solution. Therefore, the case with dual-mode control shows slightly better closed-loop cost than the case without dual-mode control in **Figure 2.4**. The graph in **Figure 2.4** quantitatively shows that the proposed interpolated solution based MBMPC can efficiently improve the closed-loop optimality compare to the existing previous solution based MBMPC with a slight additional computational cost due to the interpolation parameter.

2.5.3 Example 3 (Performance in tracking problem)

The objective function of the reference tracking problem is commonly given by

$$J = (R_s(k) - Y)^T Q_{_Y} (R_s(k) - Y) + \Delta U^T Q_{_{dU}} \Delta U \qquad (2.30)$$

where $R_s(k) \in \mathbb{R}^{Nn_y}$ denotes the reference trajectory at sampling instant k, $\Delta U := [\delta u_{0|k}; \delta u_{1|k}; \cdots ; \delta u_{N-1|k}]$ denotes the future rate of input change sequence, and $Q_{_Y}$ and $Q_{_{dU}}$ are weighting matrices with diagonal form having $\{Q_y, \cdots , Q_y\}$ and $\{Q_{du}, \cdots , Q_{du}\}$ as main diagonal blocks, respectively. This objective function can be reformulated as the quadratic function of U:

$$J_y(x, U) = U^\top H_y U + 2U^\top f_y + c_y \tag{2.31}$$

$$\begin{aligned}
H_y &:= \Psi_y^\top Q_{_Y} \Psi_y + C_2^\top Q_{_{dU}} C_2 \\
f_y &:= \Psi_y^\top Q_{_Y} (\Phi_y x - R\mathrm{s}) - C_2^\top Q_{_{dU}} \mathbf{u}_0 \\
c_y &:= (R\mathrm{s} - \Phi_y x)^\top Q_{_Y} (R\mathrm{s} - \Phi_y x) + \mathbf{u}_0^\top Q_{_{dU}} \mathbf{u}_0 \\
\Phi_y &:= (I_{_N} \otimes C)\Phi, \quad \Psi_y := (I_{_N} \otimes C)\Psi
\end{aligned}$$

In the reference tracking problem, we cannot derive the infinite-horizon LQR solution when the reference trajectory changes continually. Therefore, we used the unconstrained optimal solution $U^*_{un} := -H_y^{-1} f_y$ instead of U_{LQR}. Then, the entire input sequence of the interpolated solution based move blocking in Eq. (2.14) can be reformulated as

$$\begin{aligned}
U &= \lambda \widehat{U}(k) + (1 - \lambda) U^*_{un}(k) + P\overline{\Theta} \\
&= M(k)V + U^*_{un}(k)
\end{aligned} \tag{2.32}$$

$$M(k) := [\widehat{U}(k) - U^*_{un}(k), P]$$

and the objective function in Eq. (2.32) can be reformulated as

$$J_y = V^\top \overline{H}_y V + 2V^\top \overline{f}_y + \overline{c}_y \tag{2.33}$$

$$\begin{aligned}
\overline{H}_y &:= M(k)^\top H_y M(k) \\
\overline{f}_y &:= M(k)^\top (H_y U^*_{un}(k) + f_y) \\
\overline{c}_y &:= U^*_{un}(k)^\top H_y U^*_{un}(k) + 2U^*_{un}(k)^\top f_y + c_y
\end{aligned}$$

Finally, we can construct the standard QP problem by substituting the objective function in Eq. (2.34) into **P4** in the same manner as in *Section 2.5.2*.

We consider a chemical semi-batch reactor where a exothermic series-parallel first order reactions take place. The objective is to control the reactor temperature (T), concentration of reactant A (C_A), and volume of the solution (V) by manipulating the temperature of the jacket (T_j) and the feed flow rate of the reactant B (Q_{feed}). The following equations describe the dynamics of the semi-batch reactor [33, 34]:

$$\begin{aligned}
\frac{dT}{dt} &= \frac{Q_{feed}}{V}(T_{feed} - T) - \frac{UA}{V\rho C_p}(T - T_j) - \frac{\Delta H_1}{\rho C_p} k_{10} e^{\frac{E_1}{RT}} C_A C_B \\
&\quad - \frac{\Delta H_2}{\rho C_p} k_{20} e^{\frac{E_2}{RT}} C_B C_C \\
\frac{dC_A}{dt} &= -\frac{Q_{feed}}{V} C_A - k_{10} e^{-\frac{E_1}{RT}} C_A C_B \\
\frac{dC_B}{dt} &= \frac{Q_{feed}}{V}(C_{B,feed} - C_B) - k_{10} e^{-\frac{E_1}{RT}} C_A C_B - k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dC_C}{dt} &= -\frac{Q_{feed}}{V} C_C + k_{10} e^{-\frac{E_1}{RT}} C_A C_B - k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dV}{dt} &= Q_{feed}
\end{aligned}$$

We consider T, C_A, C_B, C_C, V as the state and assume that all states are measurable. We used the following parameters in this system:

$$T_{feed} = 308 \text{ K}, \quad C_{B,feed} = 0.9 \text{ mol/L}, \quad UA/\rho C_p = 3.75 \text{ L/min}$$
$$k_{10} = 5.0969 \times 10^{16} \text{ L/mol} \cdot \text{min}, \quad k_{20} = 2.2391 \times 10^{17} \text{ L/mol} \cdot \text{min}$$
$$E_1/R = 12305 \text{ K}, \quad E_2/R = 13450 \text{ K}, \quad \Delta H_1/(\rho C_p) = -28.5 \text{ K} \cdot \text{L/mol},$$
$$\Delta H_2/(\rho C_p) = -20.5 \text{ K} \cdot \text{L/mol}$$

The linearised discrete time model is derived with sampling instant of 1min at the initial point:

$$x_0 = [298.15; 1; 0; 0; 50], \quad u_0 = [0; 298.15]$$

The operational constraints are given by

$$0 \leq C_A \leq 1, \quad 0 \leq C_B \leq 1, \quad 0 \leq C_C \leq 1$$
$$290 \leq T \leq 310, \quad 290 \leq T_j \leq 310$$
$$50 \leq V \leq 100, \quad 0 \leq Q_{feed} \leq 0.3$$

We used the objective function in Eq. (2.32) with the control parameters of $Q_y = diag\{5, 5000, 1\}$ and $Q_{du} = diag\{10, 0.01\}$; and the prediction horizon of $N = 24$ and number of blocks of $\overline{N} = 3$; and the time-invariant blocking structure $s = \{1, 12, 24\}$.

Figure 2.5 shows the trajectories of variables in the reference tracking control of nominal MPC, and previous solution and interpolated solution based MBMPC. In the plot of controlled variable T, the proposed interpolated solution based MBMPC shows better

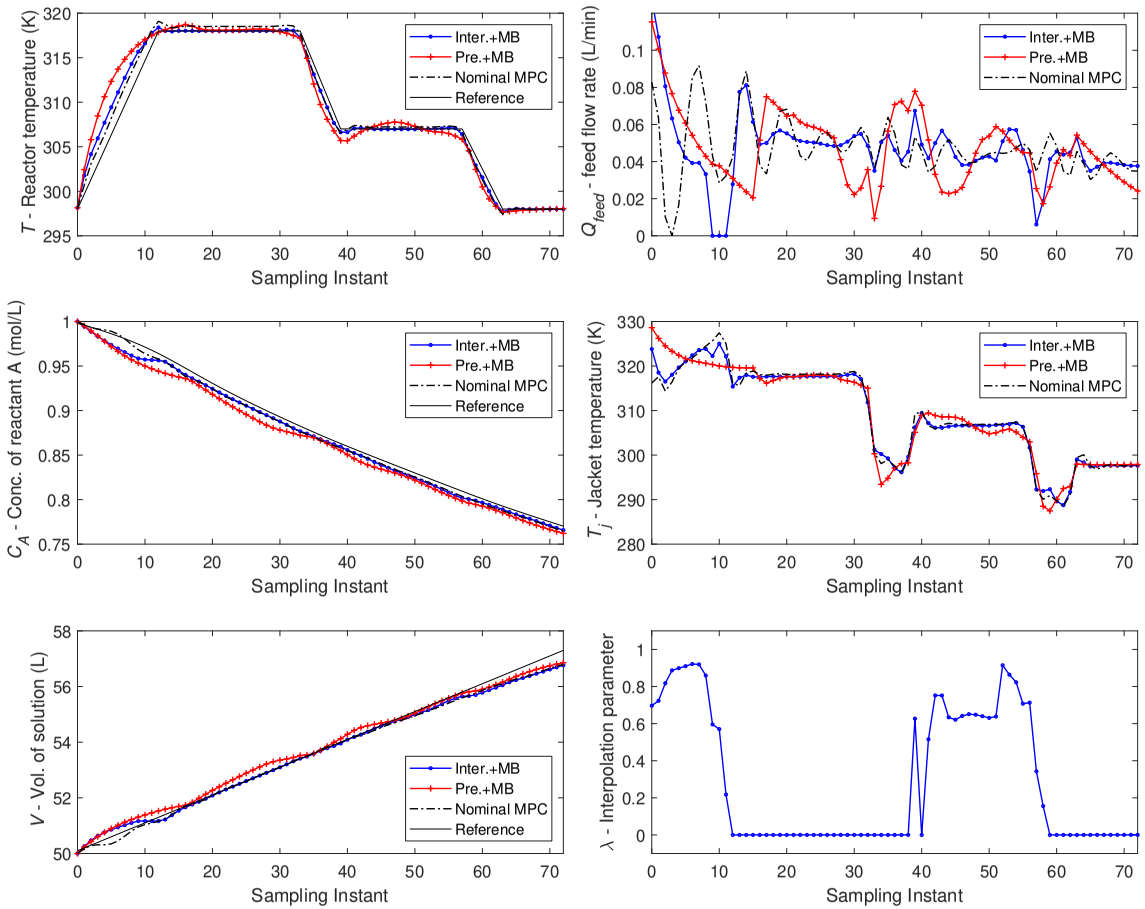

Figure 2.5: Comparison of the closed-loop performance of nominal MPC and MBMPC based on the interpolated solution, and previous solution in reference tracking problem in the chemical reactor system.

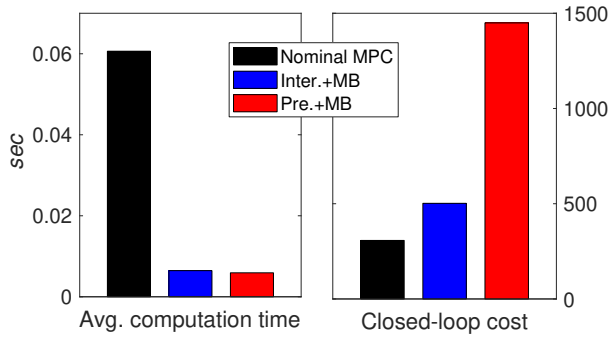

Figure 2.6: Comparison of the average computation time and closed-loop cost of nominal MPC and MBMPC based on the interpolated solution, and previous solution in reference tracking problem in the chemical reactor system.

tracking performance than that of the existing previous solution based MBMPC, while the proposed scheme shows quite similar trajectory in the plot of manipulated variable T_j. On the other hand, in case of C_A and V trajectory, the MBMPCs and nominal MPC do not show much difference, because this variable has a relatively small effect on the objective value. We can quantitatively confirm that the proposed scheme efficiently improved the closed-loop cost compared to the existing scheme with a slight additional computation time due to the interpolation parameter in **Figure 2.6**. Additionally, we can see the interpolation parameter λ does not remain as 0 after a specific sampling instant unlike the result in **Figure 2.3**. This is because, since no fixed stabilizing feedback law is available in this case owing to the continually changing reference trajectory as in the plot of controlled variables, the interpolated solution based MBMPC utilises the unconstrained optimal solution U^*_{un} instead of U_{LQR} so that the MBMPC cannot apply the concept of dual-mode control.

Chapter 3

Move-blocked model predictive control with time-varying blocking structure by semi-explicit approach[2]

3.1 Introduction

On-line computational load is one of the most outstanding issues in MPC which calculates the optimal solution for every sampling instant. Therefore, various computational complexity reduction techniques have been developed. Some techniques exploit the structure of the optimal control problem such as hierarchical decomposition approach [9], warm-starting [10], and fast MPC [11]. Other techniques reduce the order of the problem by input parameterization [12] or move blocking which fixes decision variables over arbitrary time intervals, so-called blocks [14].

Since move blocking fixes the value of decision variables in each block, the commonly used methods for ensuring recursive feasibility and closed-loop stability in MPC cannot work in move blocked MPC due to the extra constraints imposed on the control input se-

[2]This chapter is a slightly adapted version of Son, S. H., Oh, T. H., Kim, J. W., and Lee, J. M. (2019). "Move blocked model predictive control with improved optimality using semi-explicit approach for applying time-varying blocking structure". Journal of Process Control, under review.

quence. Therefore, the offset blocking scheme which fixes the deviation from the specific base sequence is widely used to guarantee the recursive feasibility and closed-loop stability by utilizing the valuable properties from the base sequence. In offset blocked MPC, the base sequence and the blocking structure act as the initial point and the search direction, respectively, in solution space. Therefore, the optimality performance of move blocked MPC is considerably affected by the optimality of the base sequence and the blocking structure. There exist several sudies based on various kinds of base sequence [14, 24]. However, since the optimal control problem of the move blocked MPC is a mixed integer program (MIP) where blocking positions are integer variables, considering the optimality of blocking structure is not a simple problem.

Some studies consider the optimality of the blocking structure in various perspectives, but the scheme which derives a proper time-varying blocking structure according to the current state on-line has not been studied yet. Shekhar and Maciejowski (2012) [31] propose a move blocking scheme which derives the time-varying optimal blocking structure by solving all the optimization problems for every admissible blocking structure on-line using parallel computing. However, this scheme cannot reduce the actual on-line computational load of the controller. Shekhar and Manzie (2015) [25] propose the move blocking scheme which derives the time-invariant optimal blocking structure in terms of maximizing the region of attractions not the optimality.

Since the on-line computation for the optimal blocking structure in MIP of move blocked MPC is prohibitive, we employ the methodology of explicit MPC [35], which moves the computational

effort for on-line optimization to off-line by exploiting the encoded state dependent information from parametric programming. Parametric programming studies the behavior of the optimizer and the value function according to the parameter (or current state) and subdivide the parameter space into several characteristic regions to depict the corresponding performance as a function of the parameter off-line [36].

In nominal MPC, the value function and optimizer function can be explicitly derived as a function of the current state when the active constraints at the optimum are known [37]. However, in case of move blocked MPC, both the active constraints and the blocking position set at the optimum are needed to specify the optimizer and value function. Therefore, when we consider all the combinations of admissible blocking position sets and the active constraint sets, the number of critical regions become exorbitant. Thus, we propose a semi-explicit approach which combines the explicit approach with simplified on-line optimization as in [38, 39, 40]. In the proposed semi-explicit move blocked MPC, we solve the multiparametric program and generate critical regions only for the blocking position set off-line. Then, we can explicitly obtain the proper time-varying blocking structure according to the current parameter by searching the critical regions, and derive the optimal blocked offset by on-line optimization of the reduced optimal control problem.

The rest of the chapter is organized as follows. We provide the standard MPC formulation with quadratic stage-wise cost in a discrete linear time-invariant system in *Section 3.2*. In *Section 3.3*, we introduce the move blocking scheme and implement it on MPC. In *Section 3.4*, we propose the semi-explicit approach for move blocked

MPC, and provide the formulation that generates the critical regions off-line and search them on-line. We also show the closed-loop property of the proposed scheme. Finally, *Section 3.5* presents the numerical examples to verify the efficacy of the proposed semi-explicit move blocked MPC scheme compared to existing methods.

3.2 Problem formulation

We consider the discrete linear time-invariant system in Eq. (3.1)

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{3.1}$$

with constraints

$$u \in \mathcal{U},\ x \in \mathcal{X} \tag{3.2}$$

where $u(k) \in \mathbb{R}^{n_u}$, $y(k) \in \mathbb{R}^{n_y}$, and $x(k) \in \mathbb{R}^{n_x}$ denote the input, output, and state, respectively. $\mathcal{U}$ and $\mathcal{X}$ are the input and state constraint sets presented as compact polyhedral region containing the origin in their interiors. We assume the pair (A, B) is stabilizable and the pair (C, A) detectable.

To render the state of the system to the origin from a given initial state, the objective function is commonly described as a sum of

quadratic stage costs as Eq. (3.3).

$$J(x_0, U) = ||x_N||^2_{Q_x^N} + \sum_{i=0}^{N-1} (||x_i||^2_{Q_x} + ||u_i||^2_{Q_u}) \tag{3.3}$$
$$U := [u_0; u_1; \cdots; u_{N-1}]$$

where $Q_u \in \mathbb{R}^{n_u \times n_u}$, $Q_x \in \mathbb{R}^{n_x \times n_x}$, and $Q_x^N \in \mathbb{R}^{n_x \times n_x}$ denote the positive definite weight matrix for input, state, and the terminal state, respectively. $[v_1; v_2; \cdots; v_n]$ represents the vertical concatenation $[v_1^\top, v_2^\top, \cdots, v_n^\top]^\top$.

We can reformulate J as a quadratic function of the input sequence by substituting Eq. (3.1) into Eq. (3.3):

$$\begin{aligned} J(x_0, U) &= U^\top H U + 2U^\top f + c \\ H &:= \Psi^\top Q_{_X} \Psi + Q_{_U} \\ f &:= \Psi^\top Q_{_X} \Phi x \\ c &:= x^\top (\Phi^\top Q_{_X} \Phi + Q_x) x \end{aligned} \tag{3.4}$$

where

$$\Phi := \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Psi := \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}$$
$$Q_{_X} := diag\{Q_x, \cdots, Q_x, Q_x^N\}$$
$$Q_{_U} := diag\{Q_u, \cdots, Q_u\}.$$

$diag\{Q_1, Q_2, \cdots, Q_n\}$ is a matrix with the diagonal form having $Q_1, Q_2, \cdots, Q_n$ as main diagonal blocks.

Then, model predictive controller solves the finite horizon optimal control problem **P1** for each sampling instant and apply the solution to the system in a receding horizon manner.

$$\begin{aligned} \mathbf{P1}: \ J^*(x) = \min_{U} \quad & J(x, U) \\ \text{s.t.} \quad & Fx + GU \leq h \end{aligned}$$

where $F \in \mathbb{R}^{n_c \times n_x}$, $G \in \mathbb{R}^{n_c \times N n_u}$ and $h \in \mathbb{R}^{n_c}$ are suitable matrices derived considering the constraint sets in Eq. (3.2) and the terminal state constraint set for the purpose of ensuring the recursive feasibility which is commonly chosen as a control invariant (CI) set [26].

3.3 Move blocked MPC

3.3.1 Move blocking scheme

Move blocking is a kind of input parameterization scheme which mitigates the computational complexity associated with optimization by fixing the value of decision variables over arbitrary time intervals, so-called blocks as illustrated in **Figure 3.1**. The blocking structure is determined by the blocking positions in **Definition 1**.

Definition 1. The ascending set $s := \{s_1, \cdots, s_{\overline{N}}\}$ is a blocking position set where $s_i \in \mathbb{N}_{\leq N}$ for $i = 1, \cdots, \overline{N}$ denotes each blocking position where the block begins, and $\overline{N}$ denotes the number of blocks ($\mathbb{N}_{\leq N}$ is the set of natural numbers less than or equal to N). In addi-

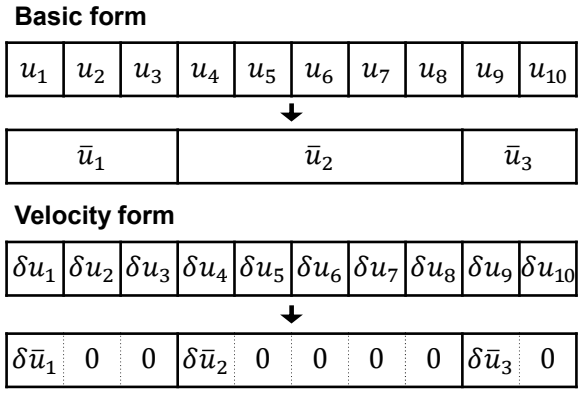

Figure 3.1: A schematic illustration of move blocking when $N = 10$, $\overline{N} = 3$, and $s = \{1, 4, 9\}$.

tion, let $\mathcal{S}$ represent the collection of admissible s.

Figure 3.1 represents the input sequence in velocity form where $\delta u_i := u_i - u_{i-1}$. The blocking matrix in velocity form is derived from s as in **Definition 2**.

Definition 2. $\overline{P}_s \in \mathbb{B}^{N\times\overline{N}}$ denotes the blocking matrix where $\mathbb{B}^{N\times\overline{N}}$ is an $N \times \overline{N}$ matrix whose elements are restricted to the binary values 0 or 1. $\overline{P}_s$ takes a lower triangular form and each column of $\overline{P}_s$ contains exactly one non-zero element. The position of non-zero elements in $\overline{P}_s$ is determined by the elements of blocking position set s:

$$\begin{aligned} &\overline{P}_s = [\overline{P}_{s_1}, \cdots, \overline{P}_{s_{\overline{N}}}] \\ &\overline{P}_{s_i} := [\mathbf{0}_{s_i-1}; 1; \mathbf{0}_{N-s_i}] \quad \text{for} \quad i = 1, \cdots, \overline{N} \end{aligned} \tag{3.5}$$

where $\mathbf{0}_m$ represents a vector of zeros with length m.

The blocking matrix in velocity form is simpler than that of the basic form, and allows for a straightforward formulation.

Remark 1. When move blocking scheme is expressed in velocity form, δu_i is non-zero only when $i \in s$, and δu_i at remaining parts are fixed as 0. From this, we can see the selection of blocking positions is identical to the selection of variables having degrees of freedom in the prediction horizon in velocity form.

Now, we can describe the parameterized input variation sequence

using the reduced variables and the blocking matrix in velocity form:

$$\Delta U = (\overline{P}_s \otimes I_{n_u})\Delta\overline{U} \tag{3.6}$$

where $\otimes$ denotes the Kronecker product, $\Delta U := [\delta u_1; \delta u_2; \cdots ; \delta u_N]$ denotes the sequence of input variation, and $\Delta\overline{U} := [\delta\overline{u}_1; \delta\overline{u}_2; \cdots ; \delta\overline{u}_{\overline{N}}]$ denotes the reduced sequence of input variation. This is the basic type of move blocking scheme called input blocking which parameterizes the input sequence by directly fixing the actual input. Input blocking scheme is simple but has limitations in terms of closed-loop performance due to the inflexibility from fixing the actual input [14].

Offset blocking fixes the offset from the base sequence not the actual input as in the input blocking:

$$\begin{cases} \Delta\Theta = (\overline{P}_s \otimes I_{n_u})\Delta\overline{\Theta} \\ \Delta U = \Delta U_b + (\overline{P}_s \otimes I_{n_u})\Delta\overline{\Theta} \end{cases} \tag{3.7}$$

where ΔU_b denotes the variation of base sequence, $\Delta\Theta := [\delta\theta_1; \delta\theta_2; \cdots ; \delta\theta_N]$ denotes the offset variation sequence, and $\Delta\overline{\Theta} := [\delta\overline{\theta}_1; \delta\overline{\theta}_2; \cdots ; \delta\overline{\theta}_{\overline{N}}]$ denotes the reduced offset variation sequence. Offset blocking scheme can deal with the limitations of input blocking scheme by exploiting the valuable properties from base sequences.

3.3.2 Implementation of move blocking

In this study, we proceed the formulation based on the offset blocking scheme in velocity form using the shifted previous solution as the base sequence with fixing the last blocking position as N,

$s_{\overline{N}} = N$, to ensure the recursive feasibility [24, 25]:

$$
\begin{aligned}
U &= \widehat{U} + C_1 P_s \Delta\overline{\Theta} \qquad (3.8)\\
\widehat{U}(k) &:= [u^*_{1|k-1}; \cdots ; u^*_{N-1|k-1}; \mathbf{0}_{n_u}]\\
C_1 &:= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \otimes I_{n_u}\\
P_s &:= \overline{P}_s \otimes I_{n_u}
\end{aligned}
$$

where $\widehat{U}$ denotes the shifted previous solution. Then, the objective function can be rewritten as a quadratic function of $\Delta\overline{\Theta}$ by substituting Eq. (3.8) into Eq. (3.4) as

$$
\begin{aligned}
J(x, \widehat{U}, s, \Delta\overline{\Theta}) &= \Delta\overline{\Theta}^\top \overline{H} \Delta\overline{\Theta} + 2\Delta\overline{\Theta}^\top \overline{f} + \overline{c} \qquad (3.9)\\
\overline{H} &:= P_s^\top C_1^\top H C_1 P_s\\
\overline{f} &:= P_s^\top C_1^\top (H\widehat{U} + f)\\
\overline{c} &:= \widehat{U}^\top H \widehat{U} + 2\widehat{U}^\top f + c.
\end{aligned}
$$

Proposition 1. $\overline{H}$ is positive definite $\forall s \in \mathcal{S}$.

Proof) $P_s^\top X P_s$ generates an $\overline{N} \times \overline{N}$ matrix consisting of $X \in \mathbb{R}^{N \times N}$ elements only corresponding to each blocking position in s. From this, we can see $P_s^\top C_1^\top H C_1 P_s$ is a principal submatrix of $C_1^\top H C_1$. Thus, since $C_1^\top H C_1$ is positive definite, $P_s^\top C_1^\top H C_1 P_s$ is also positive definite $\forall s \in \mathcal{S}$. □

When move blocking scheme is implemented on MPC, the order of existing variables is reduced by parameterization, the blocking positions are added as new variables, and additional constraints from move blocking are imposed on the optimal control problem as in **P2**.

$$\begin{aligned}\mathbf{P2}: \ J^*(x,\widehat{U}) = \min_{s,\Delta\overline{\Theta}} \ & J(x,\widehat{U},s,\Delta\overline{\Theta}) \\ \text{s.t.} \quad & \overline{F}x + \overline{G}\Delta\overline{\Theta} \leq \overline{h} \\ & s = \{s_1,\cdots,s_{\overline{N}}\}, \ s_{\overline{N}} = N \\ & s_i \in \mathbb{N}_{<N} \text{ for } i = 1,\cdots,\overline{N}-1\end{aligned}$$

As we can see above, **P2** is an MIP where blocking positions s_i are integer variables. Since the computational complexity to derive the optimal solution of MIP is too high, time-invariant blocking structures, such as control horizon [41, 42] or blocking structure which maximizes the volume of the approximate region of attraction [25], are commonly used. The computation of the time-varying optimal blocking structure by enumeration with the assumption of a parallel controller is proposed in [31, 43]. However, since this scheme cannot reduce the actual on-line computation burden, we do not consider it in this study.

3.4 Semi-explicit approach for move blocked MPC

Since the selection of blocking structure has a significant impact on optimality as described in **Remark 1**, the optimality performance of the controller can be efficiently improved when we can select a proper blocking structure given a current state on-line. However, the

on-line optimization of the MIP in **P2** is not practical. Thus, we propose to apply the explicit approach using multiparametric programming to move the on-line computational load to off-line.

In nominal MPC, the optimization problem is usually convex where the optimizer and the value function can be explicitly derived as functions of the current state by utilizing Karush-Kuhn-Tucker (KKT) conditions when the active constraints at the optimum are known. Thus, critical region in multiparametric programming of nominal MPC is usually derived for each active constraint set as in **Definition 3**.

Definition 3. Let $U^*(x)$ be the optimizer for $x \in \mathcal{X}_0$ and $I_c := \{1, \cdots, n_c\}$ be the set of constraint indices in **P1**. Then, the critical region related to a set of active constraints with the index set $\mathcal{A} \subset I_c$ is defined as

$$CR_{\mathcal{A}} = \{x \in \mathcal{X}_0 \mid F_i x + G_i U^*(x) = h_i \ \ \forall i \in \mathcal{A}\} \tag{3.10}$$

where F_i, G_i and h_i denote the i^{th} elements of F, G, and h in **P1**, respectively[36].

In case of move blocked MPC, both the active constraints and the blocking position set at the optimum are needed to specify the optimizer and value function. Therefore, when we consider all the combinations of admissible blocking position sets and the active constraint sets among the non-redundant constraints depending on each blocking position set, the number of critical regions in the worst case

can be $n_{s,\mathcal{A}}$ in Eq. (3.11).

$$n_{s,\mathcal{A}} = \sum_{i=1}^{n_s} 2^{n_{\overline{\mathcal{A}}(i)}} \tag{3.11}$$
$$n_s := \binom{N-1}{\overline{N}-1}$$

where n_s denotes the number of admissible blocking position sets with $s_{\overline{N}} = N$, and $n_{\overline{\mathcal{A}}(i)}$ denotes the number of the non-redundant constraints when the blocking structure with the i^{th} admissible blocking position set is implemented.

To move all the on-line computational load for solving **P2** to off-line, we have to investigate $n_{s,\mathcal{A}}$ combinations in Eq. (3.11). However, in this case, the off-line computational cost for generating the critical regions is significantly large, and the on-line computational cost for finding the critical region of the current parameter belongs to the parameter space would also be large. Therefore, this multiparamatric programming in fully explicit manner is not practical.

To address this limitation of move blocked MPC, we propose to proceed multiparametric programming in semi-explicit manner which generates critical regions only for the blocking position sets off-line not for the blocked offset variation sequence. Then, we can explicitly derive the proper time-varying blocking position set $\widetilde{s}^*$ for a current parameter with critical region search and solve the reduced problem on-line only for $\Delta\overline{\Theta}$ in **P3** with $\widetilde{s}^*$.

$$\begin{aligned} \mathbf{P3}: \ & J^*(x,\widehat{U},\widetilde{s}^*) = \min_{\Delta\overline{\Theta}} \ J(x,\widehat{U},\widetilde{s}^*,\Delta\overline{\Theta}) \\ & \text{s.t.} \quad \overline{F}x + \overline{G}\Delta\overline{\Theta} \leq \overline{h} \end{aligned}$$

Since the MIP in **P2** is converted to a convex optimization problem in **P3** by deriving the proper integer variables (i.e., blocking position set) explicitly, the proposed scheme considerably reduces the on-line computational complexity and makes it possible to exploit the proper time-varying blocking structure for the current parameter on-line.

3.4.1 Off-line generation of critical region

In **P1** of nominal MPC, the optimal solution depends on the current state x only, therefore, the parameter vector of multiparametric programming is identical to x. In **P2** of move blocked MPC, since the optimal solution depends on x and $\widehat{U}$, the parameter vector should be dependent on (x, $\widehat{U}$) pair. Then, the critical regions are generated in the space of the parameter vector.

Now, based on **Definition 3**, we can simply think of the critical regions for each admissible s as the sets of parameters for which the same s is optimal. To generate these critical regions, we should be able to obtain the optimal $\Delta\overline{\Theta}_s^*$ from given s. However, we cannot obtain $\Delta\overline{\Theta}_s^*$ only from the blocking structure. Therefore, we utilize the unconstrained optimal solution $\Delta\widetilde{\Theta}_s^*$ which can be easily derived from the blocking structure instead of the exact solution $\Delta\overline{\Theta}_s^*$:

$$\Delta\widetilde{\Theta}_s^*(x,\widehat{U}) = -\overline{H}^{-1}\overline{f}. \tag{3.12}$$

This allows for a straightforward formulation of the unconstrained value function $\widetilde{J}_s^*(x,\widehat{U})$ given s by substituting Eq. (3.12) into Eq.

(3.9):

$$\begin{aligned}\widetilde{J}_s^*(x,\widehat{U}) &= -\overline{f}^\top \overline{H}^{-1}\overline{f} + \overline{c} \\ &= -(H\widehat{U}+f)^\top C_1 H_s C_1^\top (H\widehat{U}+f) + \overline{c} \\ H_s &:= P_s(P_s^\top C_1^\top H C_1 P_s)^{-1} P_s^\top.\end{aligned} \tag{3.13}$$

Now, we define the parameter vector g dependent on $(x, \widehat{U})$ pair as

$$g(x,\widehat{U}) := C_1^\top (H\widehat{U}+f). \tag{3.14}$$

Then, we can reformulate $\widetilde{J}_s^*$ with g by substituting Eq. (3.14) into Eq. (3.13):

$$\widetilde{J}_s^*(g) = -g^\top H_s g + \overline{c}. \tag{3.15}$$

$\widetilde{J}_s^*(g)$ implies the unconstrained value function for the current parameter g given the blocking position set s. Let $\mathcal{S}_N$ represent the collection of admissible s with $s_{\overline{N}} = N$. Then, we can derive the minimal unconstrained value function among $\widetilde{J}_s^*(g)$ for each admissible $s \in \mathcal{S}_N$ and the minimizing blocking position set $\widetilde{s}^*$ for the current parameter g:

$$\widetilde{s}^*(g) = \operatorname{argmin}_{s\in\mathcal{S}} \ \widetilde{J}_s^*(g). \tag{3.16}$$

Let $\mathcal{I}_s := \{1, \cdots, n_s\}$ denote the set of indices for each $s \in \mathcal{S}_N$ and $s(i)$ be the blocking position set corresponding to the index $i \in \mathcal{I}_s$. Then, we define the critical regions as the sets of parameters g for

which the same blocking position set s achieves the minimal $\widetilde{J}_s^*(g)$ as in **Definition 4**.

Definition 4. Consider an admissible blocking position set $s(i) \in \mathcal{S}_N,\ i \in \mathcal{I}_s$. The critical region associated with $s(i)$ is defined as

$$CR_{s(i)} = \{g \mid \widetilde{s}^*(g) = s(i)\}. \tag{3.17}$$

Let $i^*(g) \in \mathcal{I}_s$ denote the corresponding index of $\widetilde{s}^*(g)$, i.e., $s(i^*(g)) = \widetilde{s}^*(g)$. Then, the inequality in Eq. (3.18) holds based on the definition of $\widetilde{J}_s^*(g)$ and $\widetilde{s}^*(g)$ in Eqs. (3.15) and (3.16).

$$g^\top (H_{s(i^*(g))} - H_{s(j)})g > 0 \quad \forall j \in \mathcal{I}_s \setminus i^* \tag{3.18}$$

where $H_{s(i)} := P_{s(i)}(P_{s(i)}^\top C_1^\top H C_1 P_{s(i)})^{-1} P_{s(i)}^\top$. Then, we can derive $CR_{s(i)}$ as Eq. (3.19) with Eq. (3.18) according to **Definition 4**.

$$CR_{s(i)} = \{g \mid g^\top H_{s(i,j)} g > 0 \quad \forall j \in \mathcal{I}_s \setminus i\} \tag{3.19}$$

where $H_{s(i,j)} := H_{s(i)} - H_{s(j)}$.

Proposition 2. $H_{s(i)}$ is positive semi-definite $\forall i \in \mathcal{I}_s$.

Proof) From Eq. (3.5), $P_{s(i)}\overline{X}P_{s(i)}^\top$ becomes a sparse matrix where the $\overline{X}$ elements are located on each blocking position in $s(i)$. Thus, the principle minors of $P_{s(i)}(P_{s(i)}^\top C_1^\top H C_1 P_{s(i)})^{-1} P_{s(i)}^\top$ are identical to the principle minors of $(P_{s(i)}^\top C_1^\top H C_1 P_{s(i)})^{-1}$ or zero. Since $P_{s(i)}^\top C_1^\top H C_1 P_{s(i)}$

is positive definite from **Proposition 1**, its inverse $(P_{s(i)}^{\top} C_1^{\top} H C_1 P_{s(i)})^{-1}$ is also positive definite $\forall i \in \mathcal{I}_s$. Therefore, we can see $P_{s(i)}(P_{s(i)}^{\top} C_1^{\top} H C_1 P_{s(i)})^{-1} P_{s(i)}^{\top}$ is positive semi-definite $\forall i \in \mathcal{I}_s$. □

Proposition 3. $H_{s(i,j)}$ is indefinite $\forall i, j \in \mathcal{I}_s, i \neq j$.

Proof) The position of nonzero diagonal components of $H_{s(i)}$ and $H_{s(j)}$ are identical to the elements of $s(i)$ and $s(j)$, respectively. Therefore, there is at least one nonzero diagonal component in each of $H_{s(i)}$ and $H_{s(j)}$ whose positions do not overlap with each other when $i \neq j$. We can see these nonzero diagonal components are positive from **Proposition 2**. Then, the diagonal components of $H_{s(i)} - H_{s(j)}$ corresponding to the nonoverlapped components of $H_{s(i)}$ are positive and those of $H_{s(j)}$ are negative. Thus, we can see $H_{s(i,j)}$ is indefinite $\forall i, j \in \mathcal{I}_s, i \neq j$. □

From **Proposition 3**, we can see $H_{s(i,j)}$ is not positive definite and the inequality $g^{\top} H_{s(i,j)} g > 0$ does not hold $\forall g \in \mathbb{R}^{N n_u}$. Therefore, the critical region defined in **Definition 4** is always a partition of parameter space of g, and we can subdivide the parameter space into n_s number of critical regions for each admissible blocking position set $s \in \mathcal{S}_N$ off-line. Then, we can obtain $\widetilde{s}^*$ for current parameter $g(k)$ on-line only by determining which critical region the parameter belongs to.

3.4.2 On-line MPC scheme with critical region search

In this section, we present the on-line move blocked MPC scheme with the semi-explicit approach. As the receding horizon control proceeds, the critical region where the current parameter belongs to changes according to the change of x and $\widehat{U}$. Therefore, we have to track the transition of critical region in the parameter space continually. This study utilizes an efficient line-search based point location technique presented in [44], which tracks the boundaries of critical regions where the parameter point crosses on the straight line from the previous parameter $g(k-1)$ to the current parameter $g(k)$, to track the transition of critical region.

Based on Eq. (3.19), the boundary between the critical regions $CR_{s(i)}$ and $CR_{s(j)}$ can be described as

$$B_{s(i,j)} = \{g \mid g^\top H_{s(i,j)} g = 0\}. \tag{3.20}$$

Each critical region shares $n_s - 1$ number of boundaries with other critical regions for each admissible (i, j) pair.

Now, let $i(k) \in \mathcal{I}_s$ denote the index of critical region where the current parameter $g(k)$ belongs to. Then, we can derive $i(k)$ by tracking the transition of critical region from $i(k-1)$ through successive iteration of line-search as described in **Algorithm 3.1**. $\ell_{i,j}$ denotes the scaled distance from the parameter g_0 to the boundary $B_{s(i,j)}$ along the Δg direction derived from the quadratic formula.

As shown in **Proposition 3**, $H_{s(i,j)}$ is a sparse matrix in which only the elements at positions corresponding to the components of $s(i)$ and $s(j)$ are nonzero. Therefore, only the components of parameter $g \in B_{s(i,j)}$ corresponding to $s(i) \cup s(j)$ are constrained whereas

Algorithm 3.1. Tracking of critical region transition

Initialize $g_0 = g(k-1)$, $g_f = g(k)$, and $i = i(k-1)$
loop
  Set $\Delta g = g_f - g_0$
  Perform the line search: $\ell(j^*) = \min_{j \in \mathcal{I}_s \setminus i} \ \ell_{i,j} > 0$
  $(\ell_{i,j} := \dfrac{-g_0^\top H_{i,j} \Delta g \pm (\Delta g^\top H_{i,j}(g_0 g_f^\top - g_f g_0^\top) H_{i,j} g_0)^{1/2}}{\Delta g^\top H_{i,j} \Delta g})$
  if $\ell < 1$ **then**
    Update $g_0 \leftarrow g_0 + \ell \Delta g, \ i \leftarrow j^*$
  else
    Set $i(k) = i$ and apply $i(k)$ to the controller
    Exit the loop
  end if
end loop

Table 3.1: Tracking of critical region transition

the remaining components have degrees of freedom. For example, when $N = 10$, $s(i) = \{1, 2, 3, 10\}$, and $s(j) = \{1, 4, 5, 10\}$, then the 6, 7, 8, and 9^{th} components of g have degrees of freedom. Therefore, the more number of $s(j)$ elements are identical to the elements of $s(i)$, the more sparse the $H_{s(i,j)}$ is, $g \in B_{s(i,j)}$ has more degrees of freedom, and $B_{s(i,j)}$ occupies more volume among the boundary of $CR_{s(i)}$.

In this point of view, the most dominant boundaries for $CR_{s(i)}$ are $B_{s(i,j)}$ where the components of $s(i)$ and $s(j)$ are identical except one element. Then, the set of indices j that form the dominant boundaries for $CR_{s(i)}$ can be defined as

$$\mathcal{I}_{s(i)}^{dom} = \{j \in \mathcal{I}_s \setminus i \mid n(s(i) \cap s(j)) = \overline{N} - 1\}. \tag{3.21}$$

The number of components of $\mathcal{I}_{s(i)}^{dom}$ with fixing $s(j)_{\overline{N}} = N$ to ensure

the recursive feasibility is

$$n_s^{dom} = (N - \overline{N})(\overline{N} - 1). \tag{3.22}$$

Since the number of dominant boundaries n_s^{dom} is much smaller than the number of overall boundaries $n_s - 1$ of $CR_{s(i)}$, we can efficiently reduce the computation time for line search in **Algorithm 3.1** by considering the dominant boundaries only, i.e. performing the line search for $j \in \mathcal{I}_{s(i)}^{dom}$ instead of $j \in \mathcal{I}_s \setminus i$.

By tracking the critical region where the current parameter belongs to with **Algorithm 3.1**, we obtain the proper blocking position set $\widetilde{s}^*$. Then, we obtain optimal blocked offset $\Delta\overline{\Theta}^*$ by solving the reduced problem **P3** with $\widetilde{s}^*$. The outline of move blocked MPC with semi-explicit approach is summarized in **Algorithm 3.2**.

3.4.3 Property of semi-explicit move blocked MPC

Since the shifted previous solution $\widehat{U}(k) := [u^*_{1|k-1}; \cdots ; u^*_{N-1|k-1}; \mathbf{0}_{n_u}]$ is guaranteed to be feasible until the $k + N - 1^{th}$ sampling instant, the offset blocking with $\widehat{U}$ can utilize this feasible property. Thus, the proposed semi-explicit move blocked MPC based on this offset blocking scheme can guarantee the recursive feasibility with any admissible blocking structure under the condition of $s_{\overline{N}} = N$ and $\mathcal{X}_T = \mathcal{C}_\infty$, where $\mathcal{C}_\infty$ is the maximal controlled invariant set defined below.

Definition 5. The maximal controlled invariant (MCI) set $\mathcal{C}_\infty$ is a CI

Algorithm 3.2. Semi-explicit move blocked MPC

Initialize $x(0) = x_0$, $\widehat{U}(0) = \widehat{U}_0$, $g(0) = g_0$, and $i(0) = i_0$
Set $\widetilde{s}^* = s(i(0))$ and derive $\Delta\overline{\Theta}^*$ by solving **P3**
Compute $U^* = \widehat{U}(0) + C_1 P_{s(i(0))} \Delta\overline{\Theta}^*$
Apply u_0^* to the system
Update $\widehat{U}(1) = [u_1^*; \cdots; u_{N-1}^*; \mathbf{0}_{n_u}]$
Wait for the next sampling instant 1
for $k = 1, \cdots, K$ **do**
  Measure(or estimate) $x(k)$ at sampling instant k
  Compute $g(k) = C_1^\top (H\widehat{U}(k) + f)$
  if $\Delta g(k) < \varepsilon_g$ **then**
    Set $i(k) = i(k-1)$
  else
    Obtain $i(k)$ through **Algorithm 3.1** with $j \in \mathcal{I}^{dom}_{s(i(k-1))}$
  end if
  Set $\widetilde{s}^* = s(i(k))$ and derive $\Delta\overline{\Theta}^*$ by solving **P3**
  Compute $U^* = \widehat{U}(k) + C_1 P_{s(i(k))} \Delta\overline{\Theta}^*$
  Apply u_0^* to the system
  Update $\widehat{U}(k+1) = [u_1^*; \cdots; u_{N-1}^*; \mathbf{0}_{n_u}]$
  Wait for the next sampling instant $k+1$
end for

Table 3.2: Semi-explicit move blocked MPC

set which contains all other CI sets and can also be defined as

$$\mathcal{C}_\infty := \{x_0 \in \mathcal{X} \mid \exists \{u_i \in \mathbb{R}^{n_u}\}_{i=0}^{\infty} \text{ such that } u_i \in \mathcal{U}, \\ x_{i+1} = Ax_i + Bu_i \in \mathcal{X} \ \ \forall i \in \mathbb{Z}_{\geq 0}\} \tag{3.23}$$

where $\mathbb{Z}_{\geq 0}$ is the set of non-negative integers [29].

The recursive feasibility of semi-explicit move blocked MPC is described below.

Theorem 1. Consider the optimal control problem **P3**. If $s^* \in \mathcal{S}_N$ and $\mathcal{X}_T = \mathcal{C}_\infty$, then a feasible solution is always guaranteed to exist.

Proof) We can choose the reduced offset variation sequence as $\Delta\overline{\Theta} = [0; \cdots; 0; \nu]$, $\nu \in \mathcal{U}$. If $s_{\overline{N}} = N$, then the entire input sequence can always be given by

$$\widetilde{U}(k) := [u^*_{1|k-1}; \cdots; u^*_{N-1|k-1}; \nu_k] \tag{3.24}$$

which results in the prediction of future state sequence as

$$\widetilde{X}(k) := [\widetilde{x}_{1|k}; \cdots; \widetilde{x}_{N-1|k}; \widetilde{x}_{N|k}]. \tag{3.25}$$

Since $[u^*_{1|k-1}; \cdots; u^*_{N-1|k-1}]$ is the shifted previous input solution, $\widetilde{x}_{N-1|k} \in \mathcal{X}_T$ is guaranteed. Moreover, there always exists $\nu_k \in \mathcal{U}$ which satisfies the terminal constraint $\widetilde{x}_{N|k} = A\widetilde{x}_{N-1|k} + B\nu_k \in \mathcal{X}_T$ owing to the invariance of $\mathcal{X}_T$. Therefore, we can ensure that a feasible solution of **P3** is always guaranteed to exist. □

From **Theorem 1**, feasibility of the initial values $x(0)$ and $\widehat{U}(0)$ guarantees the recursive feasibility of the proposed semi-explicit move blocked MPC.

The convergence of semi-explicit move blocked MPC can be proved in a similar manner shown in [33] with **Theorem 1**.

Theorem 2. Consider the closed-loop system under receding horizon control with input update by the semi-explicit move blocked MPC in **Algorithm 3.2**. The system will converge to the origin as $k \to \infty$

under the assumption given by

$$\begin{aligned}
&\min_{u\in\mathcal{U},x^+\in\mathcal{C}_\infty} \phi_{\Delta N}(x,u) \le 0, \quad \forall x \in \mathcal{C}_\infty \\
&\phi_{\Delta N}(x,u) := \phi(x,u) + x^{+\top} Q_x^N x^+ - x^\top Q_x^N x \\
&\phi(x,u) := x^\top Q_x x + u^\top Q_u u
\end{aligned} \tag{3.26}$$

where $x^+ = Ax + Bu$.

Proof) As in **Theorem 1**, we can always obtain a feasible input sequence $\widetilde{U}$ in Eq. (3.24) with suitable $\nu_k \in \mathcal{U}$. Then, we can derive the expected objective value when $\widetilde{U}$ is implemented:

$$J(x(k),\widetilde{U}(k)) = \sum_{i=0}^{N-2} \phi(\widetilde{x}_{i|k}, u^*_{i+1|k-1}) + \phi(\widetilde{x}_{N-1|k},\nu_k) + \widetilde{x}_{N|k}^\top Q_x^N \widetilde{x}_{N|k} \tag{3.27}$$

where $\widetilde{x}_{0|k} = x(k)$.

Since the optimal input sequence and resulting state sequence at the previous sampling instant are identical to $\widetilde{U}$ and $\widetilde{X}$ except the first and last components, we can reformulate Eq. (3.27) as

$$J(x(k),\widetilde{U}(k)) = J^*_{k-1} - \phi(x_{k-1},u_{k-1}) + \phi_{\Delta N}(\widetilde{x}_{N-1|k},\nu_k) \tag{3.28}$$

where $J^*_i := J^*(x(i),\widehat{U}(i),\widetilde{s}^*(i))$ denotes the optimal cost of the problem **P3** at the sampling instant i.

Since $\widetilde{U}$ is just a feasible solution sequence of **P3** with $\Delta\overline{\Theta} =$

$[0; \cdots; 0; \nu]$, the following inequality holds:

$$J_k^* \leq J(x(k), \widetilde{U}(k)). \tag{3.29}$$

Substituting Eq. (3.28) into Eq. (3.29) gives

$$J_k^* \leq J_{k-1}^* - \phi(x_{k-1}, u_{k-1}) + \phi_{\Delta N}(\widetilde{x}_{N-1|k}, \nu_k). \tag{3.30}$$

By the assumption in Eq. (3.26), Eq. (3.30) can be reformulated as

$$J_k^* - J_{k-1}^* \leq -\phi(x_{k-1}, u_{k-1}). \tag{3.31}$$

Since $J^* \geq 0$, the sequence of J^* strictly decreases over time. Now, summing both sides of Eq. (3.31) over all $k \geq 1$ gives

$$J_\infty^* - J_0^* \leq -\sum_{k=1}^{\infty} \phi(x_{k-1}, u_{k-1}). \tag{3.32}$$

Since $J_\infty^* \geq 0$, Eq. (3.32) can be rearranged as

$$\sum_{k=1}^{\infty} \phi(x_{k-1}, u_{k-1}) \leq J_0^*. \tag{3.33}$$

From the non-negativity of $\phi(x_{k-1}, u_{k-1})$ for all $k \geq 1$, Eq. (3.33) implies

$$\lim_{k \to \infty} \phi(x_k, u_k) = 0. \tag{3.34}$$

Since $\phi(x, u) > 0$ for all $x \neq \mathbf{0}_{n_x}$ and $u \neq \mathbf{0}_{n_u}$, we can see x_k and u_k converge to the origin as $k \to \infty$ from Eq. (3.34), and thus J_k^*

converges to zero. □

The stability of semi-explicit move blocked MPC can be proved by establishing that J^* is a Lyapunov function. Since the positivity and decreasing property are followed from the definition of J^* and Eq. (3.31), respectively, we only need to prove the continuity of J^* at the origin.

Lemma 1. Consider the optimal control problem **P3**. If $\widetilde{s}^* \in \mathcal{S}_N$ and $\mathcal{X}_T = \mathcal{C}_\infty$, then $\mathcal{X}_0 = \mathcal{X}_1 = \cdots = \mathcal{X}_{N-1} = \mathcal{C}_\infty$, where $\mathcal{X}_i$ is the set of states at time i for which **P3** is feasible:

$$\begin{cases} \mathcal{X}_i = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} \text{ such that } Ax + Bu \in \mathcal{X}_{i+1}\}, \\ \qquad\qquad \text{for } i = 0, \cdots, N-1 \\ \mathcal{X}_N = \mathcal{X}_T. \end{cases}$$

Proof) The problem is always feasible by **Theorem 1**. When $\mathcal{X}_{i+1}$ is control invariant, by the definition of CI set, $\mathcal{X}_{i+1} \subseteq \mathcal{X}_i$. Then, since there always exists $u \in \mathcal{U}$ such that $Ax + Bu \in \mathcal{X}_{i+1}$ for all $x \in \mathcal{X}_i$, $\mathcal{X}_i$ is also a CI set. Therefore, $\mathcal{X}_i$ is control invariant for all $i = 1, \cdots, N-1$ and $\mathcal{X}_0 \supseteq \mathcal{X}_1 \supseteq \cdots \supseteq \mathcal{X}_N$ holds. From this and the definition of MCI set $\mathcal{C}_\infty$, we can see $\mathcal{X}_0 = \mathcal{X}_1 = \cdots = \mathcal{X}_{N-1} = \mathcal{C}_\infty$ when $\mathcal{X}_T = \mathcal{C}_\infty$. □

Lemma 2. Consider the offset blocking controller with the fixed blocking position set $s_\nu = \{N\}$ yielding the same form of solution and resulting state sequence in Eqs. (3.24) and (3.25). If $X_T = \mathcal{C}_\infty$, then

Eq. (3.36) holds:

$$\nu_k^* = \underset{\nu_k \in \mathcal{U}, \widetilde{x}_{N|k} \in \mathcal{X}_T}{\operatorname{argmin}} \phi_{\Delta N}(\widetilde{x}_{N-1|k}, \nu_k). \tag{3.35}$$

Proof) Let $J_\nu^*(x, \widehat{U}, s_\nu)$ denote the optimal cost of the controller with s_ν. Then, we can describe $J_{\nu,k}^*$ as Eq. (3.37) in a similar manner with deriving Eq. (3.28):

$$J_{\nu,k}^* = J_{\nu,k-1}^* - \phi(x_{k-1}, u_{k-1}) + \phi_{\Delta N}(\widetilde{x}_{N-1|k}, \nu_k^*). \tag{3.36}$$

Since $\widetilde{x}_{N-1|k} \in \mathcal{C}_\infty$, we can reformulate Eq. (3.37) as

$$J_{\nu,k}^* = J_{\nu,k-1}^* - \phi(x_{k-1}, u_{k-1}) + \min_{\nu_k \in \mathcal{U}, \widetilde{x}_{N|k} \in \mathcal{X}_T} \phi_{\Delta N}(\widetilde{x}_{N-1|k}, \nu_k). \tag{3.37}$$

From this, we can see Eq. (3.36) holds. □

Theorem 3. Consider the optimal cost J^* of the optimal control problem **P3** with $\widetilde{s}^* \in \mathcal{S}_N$ and $\mathcal{X}_T = \mathcal{C}_\infty$. J^* is continuous at the origin under the assumption of Eq. (3.26).

Proof) Consider the controller in **Lemma 2**. Since the controller only updates the last component of $\widehat{U}$, the entire input sequence and resulting state sequence for $k \geq N-1$ can be written as

$$\widetilde{U}^*(k) = [\nu_{k-N+1}^*; \cdots; \nu_{k-1}^*; \nu_k^*] \tag{3.38}$$

$$\widetilde{X}^*(k) = [\widetilde{x}_{N|k-N+1}^*; \cdots; \widetilde{x}_{N|k-1}^*; \widetilde{x}_{N|k}^*]. \tag{3.39}$$

Then, we can describe $J^*_{\nu,k}$ as the sum of stage costs using $\widetilde{U}^*(k)$ and $\widetilde{X}^*(k)$ in Eqs. (3.39) and (3.40):

$$J^*_{\nu,k} = \widetilde{x}^*_{N|k}{}^\top Q^N_x \widetilde{x}^*_{N|k} + \sum_{i=k-N+1}^{k} \phi(\widetilde{x}^*_{N|i-1}, \nu^*_i) \tag{3.40}$$

where $\widetilde{x}^*_{N|k-N} = x_k$. Substituting $\phi_{\Delta N}$ in Eq. (3.26) into Eq. (3.41) yields

$$J^*_{\nu,k} = x_k^\top Q^N_x x_k + \sum_{i=k-N+1}^{k} \phi_{\Delta N}(\widetilde{x}^*_{N|i-1}, \nu^*_i). \tag{3.41}$$

From **Lemma 2**, Eq. (3.42) can be rewritten as

$$J^*_{\nu,k} = x_k^\top Q^N_x x_k + \sum_{i=k-N+1}^{k} \min_{\nu_i \in \mathcal{U}, \widetilde{x}_{N|i} \in \mathcal{X}_T} \phi_{\Delta N}(\widetilde{x}^*_{N|i-1}, \nu_i). \tag{3.42}$$

Since $s_\nu \in \mathcal{S}_N$ and $\mathcal{X}_T = \mathcal{C}_\infty$, we can see $\widetilde{x}^*_{N|i-1} \in \mathcal{X}_T$ for all $i = k-N+1, \cdots, k$ from **Lemma 1**. Therefore, we can have the inequality in Eq. (3.44) by applying the assumption in Eq. (3.26):

$$J^*_{\nu,k} \leq x_k^\top Q^N_x x_k \quad \forall x_k \in \mathcal{X}_T. \tag{3.43}$$

Since $s_\nu \subseteq \widetilde{s}^*$ when $\widetilde{s}^* \in \mathcal{S}_N$, move blocked MPC with $\widetilde{s}^*$ always provides a superior solution to that with s_ν. Therefore, it can be readily shown that

$$0 \leq J^*(x, \widehat{U}, \widetilde{s}^*) \leq J^*_\nu(x, \widehat{U}, s_\nu). \tag{3.44}$$

Applying the inequality in Eq. (3.44) to Eq. (3.45) yields

$$0 \leq J^*(x, \widehat{U}, s^*) \leq x^\top Q_x^N x \quad \forall x \in \mathcal{X}_T. \tag{3.45}$$

Since $x^\top Q_x^N x$ is continuous at the origin, $J^*(x, \widehat{U}, s^*)$ must be continuous at the origin with suitable $\widehat{U}$. □

From **Theorem 3**, we can conclude that semi-explicit move blocked MPC can steer $x \in \mathcal{X}_T$ to a level set of $J^*(x, \widehat{U}, s^*)$ contained in $\mathcal{X}_T$ where the convergence to and stability of the origin is guaranteed under the conditions of $s^* \in \mathcal{S}_N$ and $\mathcal{X}_T = \mathcal{C}_\infty$ with suitable $\widehat{U}$.

3.5 Numerical examples

In this section, we demonstrate the efficacy of the semi-explicit move blocked MPC through the examples comparing the closed-loop trajectories and on-line computation time of MPC with the time-varying blocking structure from the proposed semi-explicit approach and an arbitrary time-invariant blocking structure.

Simulations are performed using MATLAB® R2019a with Intel® Core™ i7-6700 CPU @ 3.40GHz, 32 GB RAM.

3.5.1 Example 1 (Regulation problem)

Consider the ball-plate system in [14]:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -700 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 33.18 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.7921 \end{bmatrix} u(t).$$

$x = [l_b, v_b, \phi, v_\phi]$ is the state vector where l_b and v_b denote the position and velocity of the ball, and ϕ and v_ϕ denote the angle and angular velocity of the plate, respectively. The input u is the voltage to the motor of the plate that makes the plate rotate. The operational constraints are given by

$$\begin{aligned} &-20 \leq l_b \leq 20,\ -30 \leq v_b \leq 30,\ -10 \leq \phi \leq 10 \\ &-2 \leq v_\phi \leq 2,\ -10 \leq u \leq 10. \end{aligned}$$

The discrete time model with the sampling time of $0.03s$ is derived using zero-order hold. The prediction horizon and the number of blocks are $N = 15$ and $\overline{N} = 4$, and the blocking position set $s = \{1, 6, 11, 15\}$ is used in the time-invariant blocking structure case. The initial state and base sequence are $x(0) = [5; 7; 0.1; 0.77]$ and $\widehat{U}(0) = \mathbf{0}_{15}$. The control parameters are $Q_x = diag\{6, 0.1, 500, 100\}$, $Q_{_N} = P_\infty$, and $Q_u = 1$ where P_∞ is the solution of the discrete Algebraic Riccatie Equation. The terminal MCI set was calculated using the MPT3 toolbox [32].

Figure 3.2 shows the results of MPC with semi-explicit approach, time-invariant blocking structure with $s = \{1, 6, 11, 15\}$, and non-

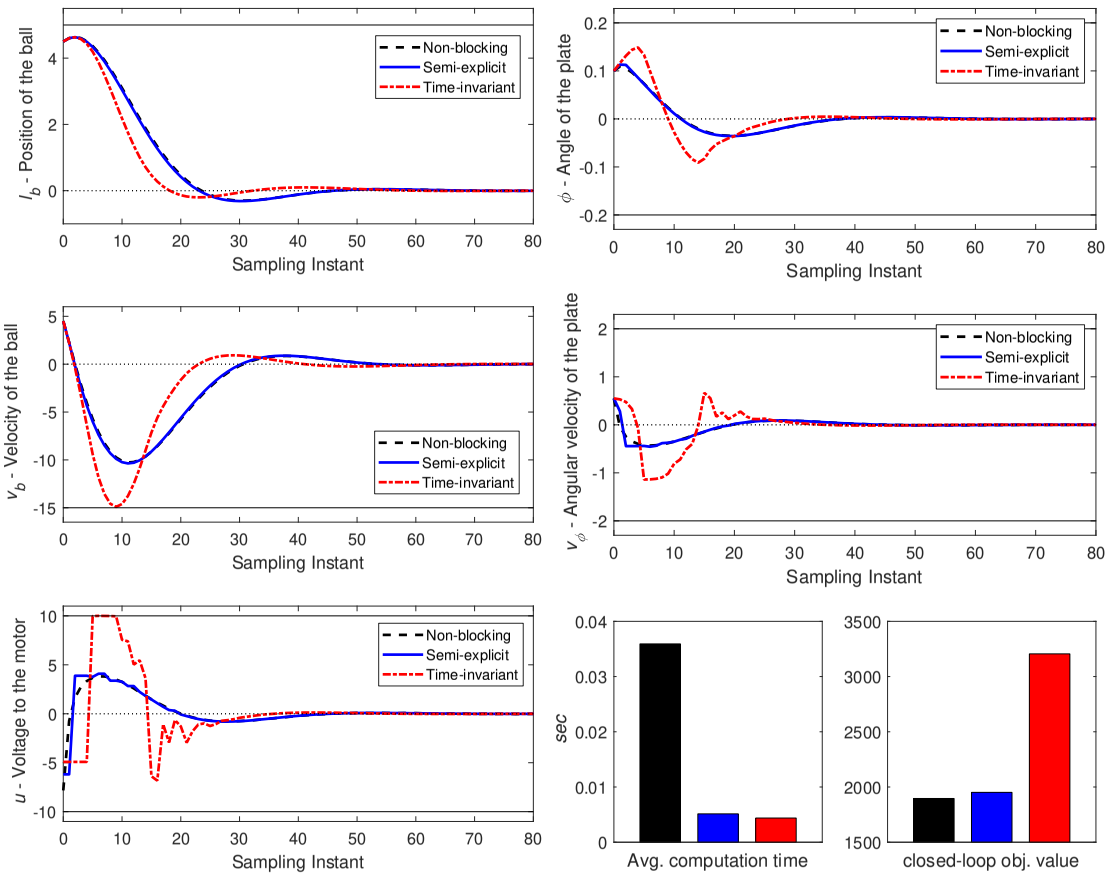

Figure 3.2: Comparison of the closed-loop trajectory, on-line computation time, and closed-loop objective value of move blocked MPC with semi-explicit approach, time-invariant blocking structure $s = \{1, 6, 11, 15\}$, and non-blocking case in the ball-plate system.

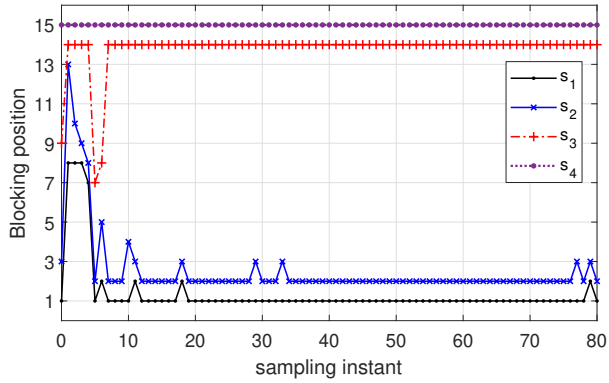

Figure 3.3: Time-varying blocking positions from the move blocked MPC with semi-explicit approach in the ball-plate system.

blocking case in a regulation problem at the ball-plate system. We can see the move blocked MPC with semi-explicit approach shows much better optimality performance than the case with time-invariant blocking structure. **Figure 3.3** illustrates the applied time-varying blocking positions for each sampling instant in move blocked MPC with semi-explicit approach. s_4 is fixed as 15 to ensure the recursive feasibility.

Since obtaining the solution by offset blocked MPC, which parameterizes the input sequence in terms of deviations from the shifted previous input solution as the base sequence, is identical with updating the retained previous input solution with a reduced number of decision variables given by the blocking structure, the selection of blocking structure has a significant influence on the optimality performance of the controller. This is why the move blocked MPC with the proposed semi-explicit approach which selects the appropriate blocking structure according to the current state shows superior performance than that with an arbitrary time-invariant blocking structure.

In addition, we can see the proposed move blocked MPC with semi-explicit approach takes slightly more amount of on-line computation time than the conventional time-invariant blocking structure case. Since both semi-explicit approach and time-invariant blocking structure case solve the optimal control problem with the same number of variables and structure, this difference in computational costs mainly comes from the critical region search in **Algorithm 3.1**. In conclusion, the proposed semi-explicit approach effectively improves the optimality performance of move blocked MPC by providing a proper blocking position set for the parameter change with compar-

atively negligible additional computation cost compared to the conventional time-invariant blocking structure case.

Since the initial condition of the system is commonly given before starting the control system, we can derive the optimal solution at the initial point $U^*(0)$ over the entire optimization window before the control starts, and then use $U^*(0)$ as the initial base sequence, $\widehat{U}(0) = U^*(0)$ and terminal cost as $Q_{_N} = 3Q_x$. At later time steps, move blocking scheme is applied. By this pre-computation technique, we can provide the fully matured initial base sequence to the controller instead of the simple zero vector $\mathbf{0}_{Nn_u}$.

Figure 3.4 shows the results of MPC with semi-explicit approach, time-invariant blocking structure with $s = \{1, 6, 11, 15\}$, and non-blocking case in regulation problem in the ball-plate system with the initial base sequence as $U^*(0)$. We can see the performance of move blocked MPC with semi-explicit approach and time-invariant blocking structure is improved compared to the results in **Figure 3.2**. This is because the optimality of the entire input sequence and the efficiency of input sequence update by offset blocking are improved by the superior initial base sequence.

Particularly, the move blocked MPC with semi-explicit approach shows similar trajectories to the nominal non-blocking case. From this, we can see the move blocking with only four blocks is sufficient to appropriately update the input sequence with the horizon of 15 in this system when the proper initial base sequence and the proper blocking structure according to the parameter change are provided through the pre-computation and the semi-explicit approach, respectively.

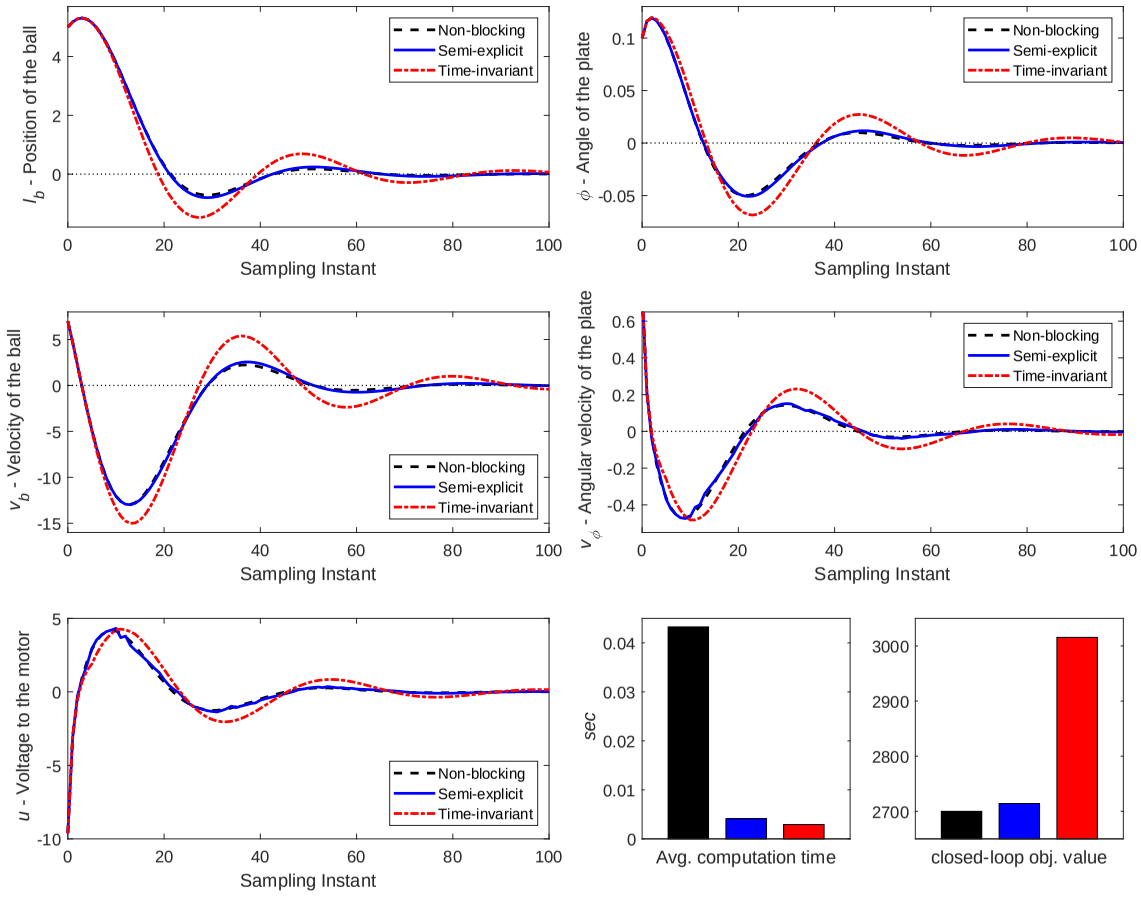

Figure 3.4: Comparison of the closed-loop trajectory, on-line computation time, and closed-loop objective value of move blocked MPC with semi-explicit approach, time-invariant blocking structure $s = \{1, 6, 11, 15\}$, and non-blocking case in the ball-plate system with the pre-computation technique.

3.5.2 Example 2 (Tracking problem)

The objective function of the tracking problem is commonly described as a sum of quadratic errors of the predicted output from the reference and input variation:

$$J(x_0, U) := \sum_{i=1}^{N} (r\mathrm{s}_i - y_i)^\top Q_y (r\mathrm{s}_i - y_i) + \delta u_i^\top Q_{du} \delta u_i \qquad (3.46)$$

where $r_{\mathrm{s}_i} \in \mathbb{R}^{n_y}$ and δu_i denote the reference signal and the input variation at sampling instant i, respectively.

The objective function in Eq. (3.47) can be rewritten as a quadratic function of the input sequence U:

$$\begin{aligned}
&J(x, U) = U^\top H_y U + 2U^\top f_y + c_y \qquad (3.47)\\
&H_y := \Psi_y^\top Q_{_Y} \Psi_y + C_2^\top Q_{_{dU}} C_2\\
&f_y := \Psi_y^\top Q_{_Y} (\Phi_y x - R\mathrm{s}) - C_2^\top Q_{_{dU}} \mathbf{u}_0\\
&c_y := (R\mathrm{s} - \Phi_y x)^\top Q_{_Y} (R\mathrm{s} - \Phi_y x) + \mathbf{u}_0^\top Q_{_{dU}} \mathbf{u}_0\\
&\Phi_y := (I_N \otimes C)\Phi, \quad \Psi_y := (I_N \otimes C)\Psi\\
&C_2 := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \otimes I_{n_u}\\
&R_s := [r_{s_1}; r_{s_2}; \cdots; r_{s_N}]\\
&\mathbf{u}_0 := [u^-; 0; \cdots; 0]
\end{aligned}$$

where u^- denotes the implemented input at the previous sampling instant.

Then, we can construct the semi-explicit move blocked MPC for the tracking problem using the same formulation in *Section 3.4* by replacing H and f with H_y and f_y.

We consider a chemical semi-batch reactor where exothermic series-parallel first order reactions take place:

$$A + B \xrightarrow{k_1} C, \quad r_1 = k_1 C_A C_B$$
$$B + C \xrightarrow{k_2} D, \quad r_2 = k_2 C_B C_C.$$

The control objective is to track the reference trajectories of the reactor temperature (T) and the concentration of reactant A (C_A) while suppressing the concentration of byproduct D (C_D) by directly manipulating the temperature of the jacket (T_j) and the feed flow rate of the reactant B (Q_{feed}). The following equations describe the dynamics of the reactor [33, 45]:

$$\begin{aligned}
\frac{dT}{dt} &= \frac{Q_{feed}}{V}(T_{feed} - T) - \frac{UA}{V\rho C_p}(T - T_j) - \frac{\Delta H_1}{\rho C_p} k_{10} e^{-\frac{E_1}{RT}} C_A C_B \\
&\quad - \frac{\Delta H_2}{\rho C_p} k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dC_A}{dt} &= -\frac{Q_{feed}}{V} C_A - k_{10} e^{-\frac{-E_1}{RT}} C_A C_B \\
\frac{dC_B}{dt} &= \frac{Q_{feed}}{V}(C_{B,feed} - C_B) - k_{10} e^{-\frac{E_1}{RT}} C_A C_B - k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dC_C}{dt} &= -\frac{Q_{feed}}{V} C_C + k_{10} e^{-\frac{E_1}{RT}} C_A C_B - k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dC_D}{dt} &= -\frac{Q_{feed}}{V} C_D + k_{20} e^{-\frac{E_2}{RT}} C_B C_C \\
\frac{dV}{dt} &= Q_{feed}.
\end{aligned}$$

We consider T, C_A, C_B, C_C, C_D, V as the state and assume that all

the states are measurable. We used the following parameters in this system:

$$
\begin{aligned}
&T_{feed} = 308 \text{ K}, \;\; C_{B,feed} = 0.9 \text{ mol/L} \\
&UA/\rho C_p = 3.75 \text{ L/min}, \;\; k_{10} = 5.0969 \times 10^{16} \text{ L/mol} \cdot \text{min}, \\
&k_{20} = 2.2391 \times 10^{17} \text{ L/mol} \cdot \text{min}, \;\; E_1/R = 12305 \text{ K}, \\
&E_2/R = 13450 \text{ K}, \;\; \Delta H_1/(\rho C_p) = -28.5 \text{ K} \cdot \text{L/mol}, \\
&\Delta H_2/(\rho C_p) = -20.5 \text{ K} \cdot \text{L/mol}.
\end{aligned}
$$

The linearized model is derived at the initial point:

$$x_0 = [298.15; 1; 0; 0; 0; 50], \quad u_0 = [0; 298.15].$$

We obtained the linear discrete time model with sampling instant of 1min. The operational constraints are given by

$$
\begin{aligned}
&0 \leq C_B \leq 0.1, \;\; 0 \leq C_C \leq 1, \;\; 0 \leq C_D \leq 0.01 \\
&293 \leq T \leq 313, \;\; 290 \leq T_j \leq 315, \;\; 0 \leq V \leq 50, \;\; 0 \leq Q_{feed} \leq 0.8.
\end{aligned}
$$

The prediction horizon and the number of blocks were $N = 24$ and $\overline{N} = 3$, and the blocking position set $s = \{1, 12, 24\}$ was used in the time-invariant blocking structure case. The initial base sequence was set as $\widehat{U}(0) = U^*(0)$ from the pre-computation. The control parameters were $Q_y = diag\{10, 700, 1\}$ and $Q_{du} = diag\{1, 0.01\}$. The terminal MCI set was calculated using the MPT3 toolbox [32].

Figure 3.5 shows the results of MPC with the semi-explicit approach, time-invariant blocking structure with $s = \{1, 12, 24\}$, and non-blocking case in a tracking problem. **Figure 3.6** illustrates the

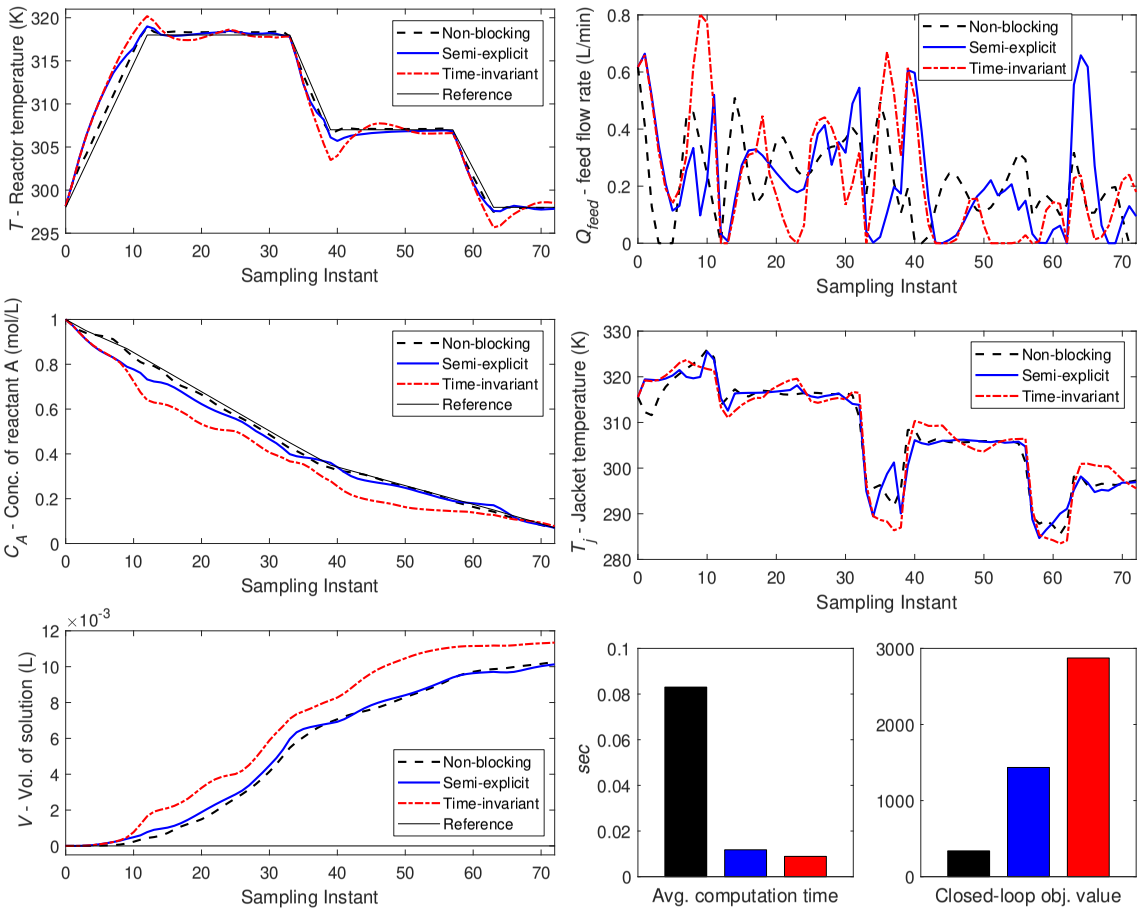

Figure 3.5: Comparison of the closed-loop trajectory, on-line computation time, and closed-loop objective value of move blocked MPC with semi-explicit approach, time-invariant blocking structure $s = \{1, 12, 24\}$, and non-blocking case in the chemical semi-batch system with the pre-computation technique.

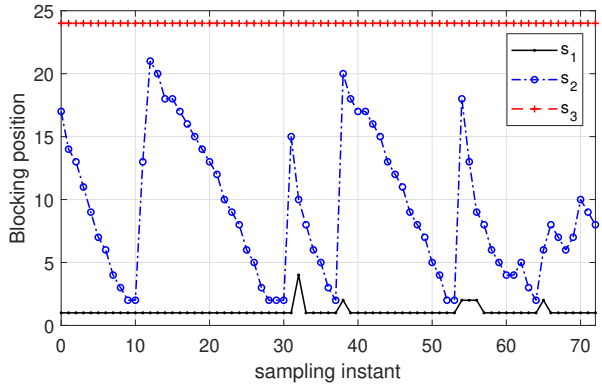

Figure 3.6: Time-varying blocking positions from the move blocked MPC with semi-explicit approach in the chemical semi-batch system.

applied time-varying blocking positions for each sampling instant in move blocked MPC with semi-explicit approach. s_3 is fixed as 24 to ensure the recursive feasibility.

The closed-loop objective values of both move blocking results with semi-explicit approach and time-invariant blocking structure show considerable disparity from that of the nominal non-blocking case. From this, we can see the move blocking with only three blocks is insufficient to appropriately update the input sequence with the horizon of 24 in this system, despite we implemented a fully matured initial base sequence on the controller by pre-computation technique. This is because the mismatch between the nonlinear dynamics of the system and the linearized model implemented on MPC leads to prediction error. Although the state of the controller is continually updated with the measured plant state, the degradation in optimality of the base sequence $\widehat{U}$ is inevitable.

Nonetheless, we can see the move blocked MPC with semi-explicit approach tracks the reference trajectory well while showing much better optimality performance than the case with time-invariant blocking structure with a slight increase in the on-line computation time. From this, we confirm that the proposed semi-explicit approach effectively improves the optimality performance of move blocked MPC in reference tracking problem.

Chapter 4

Model-plant mismatch learning offset-free model predictive control

4.1 Introduction

There are two main types of optimal control methods. Model based approach predicts the propagation of physical system with known model. Model predictive control (MPC) which derives finite horizon optimal solution in receding horizon manner is one of the most representative model-based approach. MPC effectively derives a reliable solution based on the model. Therefore, the closed-loop performance of MPC is directly related to the accuracy of model. Since model-plant mismatch and unmeasured disturbance always exist in real systems, MPC usually cannot achieve optimal performance. On the other hand, data-based machine learning (ML) approach predicts the system behavior such as dynamics and reward from the real plant data (e.g. reinforcement learning (RL) directly derives optimal control policy from data). Thus, ML does not require any given model or prior assumption about the system, and can implicitly manage uncertainties. However, this model-free pure learning approach without any prior knowledge of the system is often restricted because it requires a large amount of data and exploratory policies can damage

the system.

Since model and data based approaches are complementary to each other, combination of MPC and ML is an emerging area of research. [46, 47] derive a reliable policy by approximating a nominal MPC policy through supervised learning method such as guided policy search. [48] approximate a policy of the robust MPC by learning a neural network from robust MPC sample. [49, 50] improve MPC performance by continuously updating the dynamic model approximated with multi-layer neural network from sampled data, and [51] improve MPC performance with dynamic model updated by sparse identification of nonlinear dynamics (SINDY). [52] learns both approximate MPC policy and system dynamics with recurrent neural network and multi-layer perceptron, respectively. [53] learns a direct compensatory control action for MPC which improves the closed-loop performance of the combined controller using RL with the same performance measure of MPC.

These existing studies consider the combination of nominal MPC with ML method. However, researches for model-plant mismatch compensation have already been studied actively in MPC field to overcome the limitation of model based approach. Offset-free MPC achieves offset-free tracking in the presence of model-plant mismatch or unmeasured disturbance in two ways. One method exploited integration of tracking error in the a compensator block as in [54, 55, 56, 57]. However, since the integrated error is independent to controller, this method may cause windup problem. Therefore, disturbance estimator approach discussed in [58, 59, 60, 61, 62, 63] is the most standardly used method to accomplish offset-free tracking in MPC. This approach tracks the reference trajectory in the presence of plant-model

mismatch and unmeasured disturbances by augmenting the disturbance model to the nominal model and deriving compensatory disturbance by the estimator.

Therefore, we propose model-plant mismatch learning offset-free MPC combining data based ML and offset-free MPC. In this scheme, the general regression neural network (GRNN) proposed in [64] learns the intrinsic model-plant mismatch from the estimated disturbance data at steady-state for each set-point by supervised learning, and applies the learned model-plant mismatch into the offset-free MPC scheme. The proposed scheme also uses the nominal disturbance estimator to derive the supplementary compensating signal to exploit both learned model-plant mismatch information and stabilizing property of the nominal disturbance estimator. By this, the proposed scheme effectively improves the closed-loop performance of offset-free MPC. Moreover, we examine the robust asymptotic stability of the proposed scheme by exploiting the learned model-plant mismatch information, which has not been done in almost all offset-free MPC studies due to the difficulty in handling the combined disturbance estimator/target calculator/optimizer system ([65]).

The rest of the chapter is organized as follows. We introduce the standard offset-free MPC formulation with disturbance estimator, target calculator, and finite-horizon optimal control problem and offset-free tracking condition in *Section 4.2*. In *Section 4.3*, we propose the model-plant mismatch learning offset-free MPC scheme and examine the robust asymptotic stability of the proposed scheme. In *Section 4.4*, we present the numerical examples to demonstrate the efficiency of model-plant mismatch learning offset-free MPC compared to nominal offset-free MPC in various conditions.

4.2 Offset-free MPC: Disturbance estimator approach

We present the standard linear offset-free MPC design flow in [59]; Pannocchia and [63].

4.2.1 Preliminaries

Consider the discrete time-invariant plant in the form:

$$\begin{cases} x_p(k+1) = f_p(x_p(k), u(k)) \\ y_p(k) = g_p(x_p(k)) \\ z_p(k) = H y_p(k) \end{cases} \tag{4.1}$$

with constraints

$$u \in \mathcal{U},\ x_p \in \mathcal{X} \tag{4.2}$$

where $x_p \in \mathbb{R}^{n_{x_p}}$, $u \in \mathbb{R}^{n_u}$, $y_p \in \mathbb{R}^{n_y}$, and $z_p \in \mathbb{R}^{n_z}$ are the plant state, input, output, and controlled variables, respectively. Without loss of generality, the matrix H is assumed to have full row rank. $\mathcal{U}$ and $\mathcal{X}$ are constraint sets presented as compact polyhedral region.

The objective of offset-free linear MPC is to make the plant controlled variables z track the reference signal r with the linear time-invariant plant model in Eq. (4.3).

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{4.3}$$

where $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ are the model state and output, respec-

tively. The reference signal $r(k)$ is assumed to converge to a constant value r_∞ as $k \to \infty$. The pair (A, B) and (C, A) are assumed to be controllable and observable, respectively.

4.2.2 Disturbance estimator and controller design

The most standard method to compensate the mismatch between the plant in Eq. (4.1) and model in Eq. (4.3), and achieve offset-free reference tracking at steady-state is to augment the nominal plant model with additional integrating state so-called disturbance as in Eq. (4.4).

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + B_d d(k) \\ d(k+1) = d(k) \\ y(k) = Cx(k) + C_d d(k) \end{cases} \tag{4.4}$$

where $d \in \mathbb{R}^{n_d}$ is disturbance vector, and $B_d \in \mathbb{R}^{n_x \times n_d}$ and $C_d \in \mathbb{R}^{n_y \times n_d}$ are disturbance model matrices.

Proposition 1. The augmented system is observable if and only if the pair (C, A) is observable and

$$\text{rank} \begin{bmatrix} A - I & B_d \\ C & C_d \end{bmatrix} = n_x + n_d \tag{4.5}$$

([63]).

The appropriate matrices B_d and C_d which satisfy the condition in Eq. (4.5) can exist if and only if the number of disturbances n_d is

equal or smaller than the number of measured output n_y, $n_d \leq n_y$ ([61]).

In the assumption that B_d and C_d are chosen to satisfy the condition in **Proposition 1**, the state and disturbance estimator is designed as in Eq. (4.6).

$$\begin{aligned}\begin{bmatrix}\hat{x}(k+1)\\ \hat{d}(k+1)\end{bmatrix} &= \begin{bmatrix}A & B_d\\ 0 & I\end{bmatrix}\begin{bmatrix}\hat{x}(k)\\ \hat{d}(k)\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix}u(k)\\ &+ \begin{bmatrix}L_x\\ L_d\end{bmatrix}\left(-y_p(k) + C\hat{x}(k) + C_d\hat{d}(k)\right)\end{aligned} \tag{4.6}$$

where $L_x \in \mathbb{R}^{n_x \times n_y}$ and $L_d \in \mathbb{R}^{n_d \times n_y}$ are estimator gains for state and disturbance, respectively, chosen to make the estimator stable.

The following finite-horizon optimal control problem P_0 is solved in receding horizon manner.

$$\begin{aligned}\mathrm{P}_0:\ J_0^0(\hat{x}_d) = \min_{u_0,\cdots,u_{N-1}}\ & \phi_t(\bar{x}, x_N) + \sum_{i=0}^{N-1}\phi(\bar{x}, \bar{u}, x_i, u_i)\\ \text{s.t.}\quad & x_0 = \hat{x},\ d = \hat{d}\\ & x_{i+1} = Ax_i + Bu_i + B_d d\\ & u_i \in \mathcal{U},\ x_{i+1} \in \mathcal{X},\ x_N \in \mathcal{X}_t\\ & i = 0, \ldots, N\end{aligned}$$

with target state $\bar{x}$ and input $\bar{u}$ are derived from Eq. (4.7).

$$\begin{bmatrix}A - I & B\\ HC & 0\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{u}\end{bmatrix} = \begin{bmatrix}-B_d\hat{d}\\ r - HC_d\hat{d}\end{bmatrix} \tag{4.7}$$

where $\phi(\bar{x}, \bar{u}, x_i, u_i) := ||x_i - \bar{x}||^2_{Q_x} + ||u_i - \bar{u}||^2_{Q_u}$ and $\phi_t(\bar{x}, x_N) := ||x_N - \bar{x}||^2_{Q_x^N}$ denote the single-stage cost and terminal cost with $||x||^2_Q := x^\top Q x$, respectively, and $\mathcal{X}_t$ denotes the terminal constraint set. $Q_x \in \mathbb{R}^{n_x}$ and $Q_u \in \mathbb{R}^{n_u}$ are weighting matrices with diagonal form. $r \in \mathbb{R}^{n_z}$ denotes the reference signal which is assumed to converge to a constant, $r(k) \to r_\infty$ as $k \to \infty$.

4.2.3 Offset-free tracking condition

By rearranging Eq. (4.6), we can see the disturbance estimator satisfies Eq. (4.8) at steady-state.

$$\begin{bmatrix} A - I + L_x C & B \\ L_d C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} L_x y_{p,\infty} - (B_d + L_x C_d)\hat{d}_\infty \\ L_d y_{p,\infty} - L_d C_d \hat{d}_\infty \end{bmatrix} \tag{4.8}$$

where ∞ denotes steady-state values.

Let κ_{un} denote the unconstrained MPC controller gain of P_0. Since we assume constraints are not active at the equilibrium point, κ_{un} is optimal and feasible. Therefore, the input u_∞ and target input $\bar{u}_\infty$ at steady-state satisfies Eq. (4.9).

$$u_\infty - \bar{u}_\infty = \kappa_{un}(\hat{x}_\infty - \bar{x}_\infty). \tag{4.9}$$

Define steady-state output prediction error and controlled variable offset as in Eqs. (4.10) and (4.11).

$$e_{y,\infty} := y_{p,\infty} - C\hat{x}_\infty - C_d\hat{d}_\infty. \tag{4.10}$$

$$e_{z,\infty} := H y_{p,\infty} - r_\infty. \tag{4.11}$$

Then, the offset-free tracking condition can be represented as in **Proposition 2**.

Proposition 2. If following condition in Eq. (4.12) is satisfied, then offset-free tracking is achieved.

$$\mathcal{N}(L_d) \subseteq \mathcal{N}(H(I - C(I - A - B\kappa_{un})^{-1}L_x) \tag{4.12}$$

where $\mathcal{N}$ denotes null spaces ([63]).

Proof). By combining and rearranging Eqs. (4.7)–(4.11), we can derive Eq. (4.13).

$$\begin{bmatrix} L_d \\ H(I - C(I - A - B\kappa_{un})^{-1}L_x) \end{bmatrix} e_{y,\infty} = \begin{bmatrix} 0 \\ I \end{bmatrix} e_{z,\infty} \tag{4.13}$$

We can see that following condition in Eq. (4.14) should be satisfied to achieve offset-free tracking, i.e., $e_{z,\infty} = 0$, for all $e_{y,\infty}$ satisfying $L_d e_{y,\infty} = 0$.

$$H(I - C(I - A - B\kappa_{un})^{-1}L_x e_{y,\infty} = 0 \tag{4.14}$$

Then, this condition can be reformulated into the null space condition in Eq. (4.12). □

Remark 1. To construct the L_x and L_d satisfying the condition in Eq. (4.12), [59] suggests the following estimator gain structure in Eq.

(4.15).

$$\begin{bmatrix} L_x \\ L_d \end{bmatrix} = \begin{bmatrix} L_x^0 \\ 0 \end{bmatrix} + \begin{bmatrix} \bar{L}_x \\ \bar{L}_d \end{bmatrix} H(I - C{e_{\kappa_x,\infty}}^{-1} L_x^0) \tag{4.15}$$
$$e_{\kappa_x,\infty} := I - A - B\kappa_{un}.$$

When the number of disturbance n_d and the number of measured output n_y is identical, $n_d = n_y$, by *Proposition 2* in [59], L_d is nonsingular. In this case, $e_{y,\infty}$ is naturally become 0 at steady-state, therefore, a simple gain structure, where $L_x^0 = 0$ and $\bar{L}_x$, $\bar{L}_d$ are chosen to stabilize the estimator in Eq. (4.6), can satisfy the offset-free tracking condition in **Proposition 2**. In case of $n_d < n_y$, L_x^0, $\bar{L}_x$, and $\bar{L}_d$ can be constructed according to the procedures suggested in *Algorithm 4.2* and *Algorithm 4.3* in [59].

4.3 Model-plant mismatch learning offset-free MPC

Estimated disturbance $\hat{d}$ in Eq. (4.6) makes the predicted output from model be identical to the plant output by compensating the effect of model-plant mismatch. This compensating disturbance signal has the intrinsically given value for each state and input pair (x, u) with proper disturbance model matrices, B_d, C_d, and is naturally $\mathbf{0}_{n_d}$ when the model-plant mismatch does not exist. Therefore, we propose a perspective to regard this compensating disturbance value for each (x, u) pair as the intrinsic model-plant mismatch itself and the scheme to learn and exploit this model-plant mismatch.

Though the nominal offset-free MPC in *Section 4.2* has its own model-plant mismatch compensating property, it basically estimates

the proper disturbance from the occurred measurement error. Therefore, there exists some delay to estimate proper compensating signal. However, since the model-plant mismatch compensating property of nominal offset-free MPC is sensitive to the estimator performance, considerable performance degradation can occur in system transition such as set-point change.

To overcome this limitation, we propose to learn the intrinsic model-plant mismatch from the past estimated steady-state disturbance data and apply the learned model-plant mismatch to the disturbance estimator, target calculator, and model-based finite-horizon optimal control problem to improve the closed-loop performance of offset-free MPC.

4.3.1 Model-plant mismatch learning

If we could obtain the entire model-plant mismatch map for every state and input pair $(x, u) \in \mathbb{R}^{n_x \times n_u}$, we can derive the actual optimal solution for real plant. However, since obtaining the entire model-plant mismatch map is consequently identical to obtaining the exact entire plant dynamics, it needs massive data and computation.

Therefore, we propose to learn and utilize a reduced model-plant mismatch map only for steady-state pairs $(\hat{x}_\infty, u_\infty)$ for each set-point. This reduced model-plant mismatch map is a tiny partial manifold on the entire model-plant mismatch map for every (x, u) pair. Therefore, we can derive this reduced model-plant mismatch map from considerably smaller amount of data and computation than those for the entire map.

Lemma 1. Suppose that the estimator in Eq. (4.6) is stable. Then, the matrix
$\begin{bmatrix} A - I + L_x C & B_d + L_x C_d \\ L_d C & L_d C_d \end{bmatrix}$ is nonsingular.

Proof). Rearranging Eq. (4.6) follows Eq. (4.16).

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = \begin{bmatrix} A + L_x C & B_d + L_x C_d \\ L_d C & I + L_d C_d \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) - \begin{bmatrix} L_x \\ L_d \end{bmatrix} y_p(k) \tag{4.16}$$

Since we assumed the estimator is stable, it has no poles at (1,0). Therefore, following Eq. (4.17) is satisfied.

$$\det \left(\begin{bmatrix} A + L_x C & B_d + L_x C_d \\ L_d C & I + L_d C_d \end{bmatrix} - I \right) \neq 0 \tag{4.17}$$

Thus, the matrix $\begin{bmatrix} A - I + L_x C & B_d + L_x C_d \\ L_d C & L_d C_d \end{bmatrix}$ is nonsingular. □

With **Lemma 1**, we can show the existence of state and disturbance estimate pair at steady-state for each set-point.

Theorem 1. If a set-point $\bar{r}$ is achievable, i.e., there exist a steady-state plant output and input pair ($y_{p,\infty}$, u_∞) which achieves $\bar{r}$, then an estimated state and disturbance pair of the model ($\hat{x}_\infty$, $\hat{d}_\infty$) at that steady-state always exists.

Proof). If a set-point $\bar{r}$ is achievable, we can see Eqs. (4.18) and

(4.19) are satisfied at that steady-state from the estimator in Eq. (4.6).

$$\begin{aligned}\hat{x}_\infty &= (A + L_x C)\hat{x}_\infty + (B_d + L_x C_d)\hat{d}_\infty \\ &\quad + Bu_\infty - L_x y_{p,\infty} \qquad (4.18)\\ \hat{d}_\infty &= L_d C\hat{x}_\infty + (I + L_d C_d)\hat{d}_\infty - L_d y_{p,\infty} \qquad (4.19)\end{aligned}$$

where $y_{p,\infty}$ denotes the steady-state plant output which satisfies $\bar{r} = Hy_{p,\infty}$, u_∞ denote the steady-state input, and $\hat{x}_\infty$ and $\hat{d}_\infty$ denote the state and disturbance estimates at steady-state, respectively. By rearranging Eqs. (4.18) and (4.19), we can derive Eq. (4.20).

$$\begin{bmatrix} A - I + L_x C & B_d + L_x C_d \\ L_d C & L_d C_d \end{bmatrix} \begin{bmatrix} \bar{r} \\ \hat{d}_\infty \end{bmatrix} = \begin{bmatrix} L_x y_{p,\infty} - Bu_\infty \\ L_d y_{p,\infty} \end{bmatrix} \quad (4.20)$$

Since the matrix $\begin{bmatrix} A - I + L_x C & B_d + L_x C_d \\ L_d C & L_d C_d \end{bmatrix}$ is nonsingular from **Lemma 1**, we can see the pair ($\hat{x}_\infty$, $\hat{d}_\infty$) always exists whenever a steady-state plant output and input pair ($y_{p,\infty}$, u_∞) achieving $\bar{r}$ exists. □

By rearranging Eq. (4.20), the ($\hat{x}_\infty$, $\hat{d}_\infty$) pair can be derived directly from the ($y_{p,\infty}$, u_∞) pair.

$$\begin{bmatrix} \hat{x}_\infty \\ \hat{d}_\infty \end{bmatrix} = \begin{bmatrix} A - I + L_x C & B_d + L_x C_d \\ L_d C & L_d C_d \end{bmatrix}^{-1} \begin{bmatrix} L_x & -B \\ L_d & 0 \end{bmatrix} \begin{bmatrix} y_{p,\infty} \\ u_\infty \end{bmatrix} \quad (4.21)$$

Remark 2. We can see the ($\hat{x}_\infty$, $\hat{d}_\infty$) pair is derived by the linear

transformation from the corresponding ($y_{p,\infty}$, u_∞) pair as in Eq. (4.21). Therefore, the uniqueness of the ($\hat{x}_\infty$, $\hat{d}_\infty$) pair for each set-point is not confirmed when more than one ($y_{p,\infty}$, u_∞) pairs can achieve the set-point $\bar{r}$. In this study, we focus on the case where the only one ($y_{p,\infty}$, u_∞) pair achieves each $\bar{r}$, and thus, the uniqueness of ($\hat{x}_\infty$, $\hat{d}_\infty$) for each $\bar{r}$ is ensured.

Since the existence and uniqueness of ($\hat{x}_\infty$, $\hat{d}_\infty$) for each $\bar{r}$ are ensured by **Theorem 1** and **Remark 2**, we can define an intrinsic relation between $\bar{r}$ and $\hat{d}_\infty$ as Eq. (4.22).

$$\hat{d}_\infty = f_d(\bar{r}) \tag{4.22}$$

The function $f_d : \mathbb{R}^{n_z} \to \mathbb{R}^{n_d}$ implies the intrinsic model-plant mismatch of the system.

We approximate the unknown function f_d in Eq. (4.22) from the estimated disturbance data by GRNN. GRNN is a variation to radial basis function (RBF) based neural networks for non-parametric regression proposed by [64] which approximates the probability density function using Parzens non-parametric estimator with Gaussian activation function.

GRNN can be interpreted as a normalized RBF network with hidden units centered at every training sample. The predicted output $o(i)$ from input i by GRNN is a weighted average of outputs in the training set:

$$o(i) = \frac{\sum_{s=1}^{N_s} o_s \omega(i, i_s)}{\sum_{s=1}^{N_s} \omega(i, i_s)} \tag{4.23}$$

where N_s is the number of training samples, and $\omega(i, i_s)$ denotes the weight. Each weight is an RBF output which is the exponential of the negatively scaled distance between the new pattern and each given training pattern:

$$\omega(i, i_s) = e^{-(i-i_s)^\top (i-i_s)/2\sigma^2} \tag{4.24}$$

where σ is the smoothing factor which represents the width of RBF [66].

Remark 3. GRNN is a single-pass learning network with no training parameters while the back-propagation neural network (BPNN) needs forward and backward pass training. The only adjustable parameter in GRNN is smoothing factor ρ.

From **Remark 3**, GRNN needs significantly less time for training than BPNN. By this notable advantage on rapid training, GRNN is suitable for on-line systems or systems which require minimal computation [67].

However, since the number of the neurons in hidden layer is equal to the number of the training samples, the size of GRNN can be huge. We overcome this limitation by using only a few recent data in receding data-window manner for regression. This strategy is reasonable, since the characteristics of the system can change during operation.

4.3.2 Application of learned model-plant mismatch

Now, we can utilize the learned model-plant mismatch map $\hat{d}^{\ell}$ for each set-point $\bar{r}$ derive by GRNN in Eq. (4.25).

$$\hat{d}^{\ell} = \hat{f}_d(\bar{r}) \tag{4.25}$$

where $\hat{f}_d$ is the approximated function of f_d in Eq. (4.22). As described in *Section 4.3.1*, this reduced model-plant mismatch map is reasonably obtainable. However, since this learned model-plant mismatch map is only a tiny manifold on steady-state pairs $(\hat{x}_{\infty}, u_{\infty})$, we can figure out the target equilibrium point but cannot reach that point only with this map.

Therefore, we propose to incorporate the learned model-plant mismatch into the nominal disturbance estimator to exploit the stabilizing property and achieve the offset-free tracking property. For this, we introduced an additional supplementary disturbance $\hat{d}^{s}$ and combine it with the learned model-plant mismatch $\hat{d}^{\ell}$:

$$\hat{d}^{\ell,s} = \hat{d}^{\ell} + \hat{d}^{s}. \tag{4.26}$$

The supplementary signal $\hat{d}^{s}$ is continually updated by the estimator. The revised disturbance estimator estimates the state $\hat{x}$ and the supplementary disturbance $\hat{d}^{s}$ including the influence of learned

model-plant mismatch $\hat{d}^{\ell}$ as in Eq (27).

$$\begin{bmatrix} \hat{x}^{\ell,s}(k+1) \\ \hat{d}^{s}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}^{\ell,s}(k) \\ \hat{d}^{s}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} B_d \\ 0 \end{bmatrix} \hat{d}^{\ell}(k) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (-y_p(k) + C\hat{x}^{\ell,s}(k) + C_d(\hat{d}^{\ell}(k) + \hat{d}^{s}(k))) \tag{4.27}$$

The stability of the estimator in Eq. (4.27) can be simply proved in **Theorem 2**.

Theorem 2. If L_x and L_d are chosen to make the nominal disturbance estimator in Eq. (4.6) stable, then proposed disturbance estimator in Eq. (4.27) is also stable with the same L_x and L_d.

Proof). Rearranging Eq. (4.27) follows Eq. (4.28).

$$\begin{bmatrix} \hat{x}^{\ell,s}(k+1) \\ \hat{d}^{s}(k+1) \end{bmatrix} = \begin{bmatrix} A + L_x C & B_d + L_x C_d \\ L_d C & I + L_d C_d \end{bmatrix} \begin{bmatrix} \hat{x}^{\ell,s}(k) \\ \hat{d}^{s}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) - \begin{bmatrix} L_x \\ L_d \end{bmatrix} y_p(k) + \begin{bmatrix} B_d + L_x C_d \\ L_d C_d \end{bmatrix} \hat{d}^{\ell}(k) \tag{4.28}$$

Let x and d^s denote the exact model state and supplementary disturbance:

$$\begin{bmatrix} x(k+1) \\ d^{s}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ d^{s}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k). \tag{4.29}$$

These exact values satisfy Eq. (4.30).

$$y_p(k) = Cx(k) + C_d(d^s(k) + \hat{d}^\ell(k)). \tag{4.30}$$

By substituting Eq. (4.30) into Eq. (4.28) and subtracting Eq. (4.29) from Eq. (4.28), we obtain the error dynamics in Eq.(4.31).

$$\begin{bmatrix} e_{\hat{x}}(k+1) \\ e_{\hat{d}^s}(k+1) \end{bmatrix} = \begin{bmatrix} A + L_x C & B_d + L_x C_d \\ L_d C & I + L_d C_d \end{bmatrix} \begin{bmatrix} e_{\hat{x}}(k) \\ e_{\hat{d}^s}(k) \end{bmatrix} \tag{4.31}$$

where $e_{\hat{x}} := x - \hat{x}^{\ell,s}$ and $e_{\hat{d}^s} := d^s - \hat{d}^s$.

Since the matrix $\begin{bmatrix} A + L_x C & B_d + L_x C_d \\ L_d C & I + L_d C_d \end{bmatrix}$ is the same as that of the nominal disturbance estimator in Eq. (4.6), we can see the proposed estimator is stable with the same L_x and L_d of the nominal estimator. □

From the stability of the revised disturbance estimator proved in **Theorem 2**, we can exploit the learned model-plant mismatch while exploiting the stabilizing property of the disturbance estimator in Eq. (4.27) with the supplementary signal $\hat{d}^s$.

Remark 4. Even when the $\hat{d}^\ell$ from GRNN is not properly learned or the intrinsic model-plant relation is changed by system transformation or unknown disturbance injection, the supplementary signal $\hat{d}^s$ from disturbance estimator in Eq. (4.27) compensates the model-plant mismatch and achieve the offset-free tracking property.

Then, we apply $\hat{x}^{\ell,s}$ and $\hat{d}^{\ell,s}$ to the finite-horizon optimal control

problem P_0 and target calculation problem in Eq. (4.7) to improve the closed-performance of the offset-free MPC by improving prediction accuracy.

The offset-free tracking property of the proposed model-plant mismatch learning offset-free MPC is proved in **Theorem 3**.

Theorem 3. The same estimator gain constructed as in **Remark 1** achieves the offset-tracking property in proposed model-plant mismatch learning offset-free MPC.

Proof). By rearranging Eq. (4.27), we obtain Eqs. (4.32) and (4.33) at steady-state.

$$\hat{x}_{\infty}^{\ell,s} = A\hat{x}_{\infty}^{\ell,s} + Bu_{\infty}^{\ell,s} + B_d\hat{d}_{\infty}^{\ell,s} - L_x e_{y,\infty}^{\ell,s} \tag{4.32}$$

$$0 = L_d e_{y,\infty}^{\ell,s} \tag{4.33}$$

$$e_{y,\infty}^{\ell,s} := -y_{p,\infty}^{\ell,s} + C\hat{x}_{\infty}^{\ell,s} + C_d\hat{d}_{\infty}^{\ell,s}. \tag{4.34}$$

where $e_{y,\infty}^{\ell,s}$ denotes the output reconstruction error of $\mathrm{P}^{\ell,s}$ at steady-state. From the target calculator in Eq. (4.7), we can derive Eqs. (4.35) and (4.36).

$$\bar{x}_{\infty}^{\ell,s} = A\bar{x}_{\infty}^{\ell,s} + B\bar{u}_{\infty}^{\ell,s} + B_d\hat{d}_{\infty}^{\ell,s} \tag{4.35}$$

$$\bar{r} = HC\bar{x}_{\infty}^{\ell,s} + HC_d\hat{d}_{\infty}^{\ell,s} \tag{4.36}$$

The steady-state input can be derived from the unconstrained optimal

gain κ_{un} of P_0:

$$u_\infty^{\ell,s} - \bar{u}_\infty^{\ell,s} = \kappa_{un}(\hat{x}_\infty^{\ell,s} - \bar{x}_\infty^{\ell,s}). \tag{4.37}$$

Subtracting Eq. (4.35) from Eq. (4.32) and substituting Eq. (4.37) yields

$$\hat{x}_\infty^{\ell,s} - \bar{x}_\infty^{\ell,s} = -(I - A - B\kappa_{un})^{-1} L_x e_{y,\infty}^{\ell,s}. \tag{4.38}$$

Now, let $e_{z,\infty}^{\ell,s}$ denote the offset vector of the controlled variables at steady-state:

$$e_{z,\infty}^{\ell,s} := H y_{p,\infty}^{\ell,s} - \bar{r}. \tag{4.39}$$

Substituting Eq. (4.36) into Eq. (4.39) and rearranging yields

$$e_{z,\infty}^{\ell,s} = H(y_{p,\infty}^{\ell,s} - C\hat{x}_\infty^{\ell,s} - C\hat{d}_\infty^{\ell,s} + C(\hat{x}_\infty^{\ell,s} - \bar{x}_\infty^{\ell,s})). \tag{4.40}$$

Then, substituting Eqs. (4.34) and (4.38) into (4.40) yields

$$e_{z,\infty}^{\ell,s} = H[I - C(I - A - B\kappa_{un})^{-1} L_x] e_{y,\infty}^{\ell,s}. \tag{4.41}$$

Finally, combining and rearranging Eqs. (4.33) and (4.41), we obtain

$$\begin{bmatrix} L_d \\ H(I - C(I - A - B\kappa_{un})^{-1} L_x) \end{bmatrix} e_{y,\infty}^{\ell,s} = \begin{bmatrix} 0 \\ I \end{bmatrix} e_{z,\infty}^{\ell,s}. \tag{4.42}$$

Since Eq. (4.42) is identical to Eq. (4.13) in **Proposition 2**, the proposed scheme achieves offset-free tracking property with the same

estimator gain as that of nominal scheme. □

4.3.3 Robust asymptotic stability of model-plant mismatch learning offset-free MPC

In order to show the closed-loop asymptotic stability of offset-free MPC, we have to examine the closed-loop behavior of the combined system consisting of disturbance estimator / target calculator / optimizer. Since it is known to be difficult to examine the closed-loop behavior of the combined system, almost all offset-free MPC schemes only show the offset-free tracking property when the stable equilibrium has been reached as in **Theorem 3** ([65]).

The most crucial threshold for this problem is that the nominal offset-free MPC could not specify an equilibrium point and Lyapunov function candidate until the system reaching the steady-state, since the target state and input in P_0 are continually updated from the target calculator. However, in the proposed model-plant mismatch learning offset-free MPC scheme, we can specify the equilibrium point from the learned model-plant mismatch and provide the Lyapunov function candidate as the value function of P_0 with the fixed target as the specified equilibrium point. By this, we can handle the above-mentioned threshold of nominal offset-free MPC and examine the robust asymptotic stability of the proposed scheme.

With the specified equilibrium point and Lyapunov function candidate from the learned model-plant mismatch, we examine the closed-loop robust asymptotic stability of model-plant mismatch learning offset-free MPC based on the framework in [68] and [69] which show the closed-loop robust asymptotic stability of the combined estima-

tor/optimizer system.

[69] define the robust asymptotic stability as the input-to-state stability (ISS) stability ([70]) on a robust positive invariant set.

Definition 1. (Robust positive invariance) A set $\mathcal{O} \subseteq \mathcal{Y}$ is said to be robust positive invariant for an autonomous system $y^+ = g(y, e)$ with perturbation e if there exists some $\delta_e > 0$ such that $g(y, e) \subseteq \mathcal{O}$ for all $y \in \mathcal{O}$ and $\mathbf{e}$ satisfying $||\mathbf{e}|| \leq \delta_e$ where $||v||$ denotes the supremum norm $||v|| = \sup_{k \geq 0} |v(k)|$ and $\mathbf{e}$ denotes the sequence of e ([69]).

Definition 2. (Robust asymptotic stability) The equilibrium point $\bar{y}$ of a perturbed system $y^+ = g(y, e)$ is robust asymptotically stable in $\mathcal{O}$ if there exists some $\delta > 0$ such that for all perturbation sequence $\mathbf{e}$, $||\mathbf{e}|| < \delta$, $\mathcal{O}$ is robust invariant and there exist a class $\mathcal{KL}$ function $\beta(\cdot)$ and a class $\mathcal{K}$ function $\gamma(\cdot)$ satisfying

$$|y_k^+(y, \mathbf{e}) - \bar{y}| \leq \beta(|y - \bar{y}|, k) + \gamma(||\mathbf{e}||) \tag{4.43}$$

for each $y \in \mathcal{O}$ and for all $k \in \mathbb{I}_{\geq 0}$ where $y_k^+(y, \mathbf{e})$ is the open-loop solution of the perturbed system for given step k and $\mathbb{I}_{\geq 0}$ is the set of non-negative integers ([69]).

Then, [68] show the robust asymptotic stability of the origin of the combined system by establishing that the value function of the finite-horizon optimal control problem is an ISS Lyapunov function for the combined system and applying **Proposition 3**.

Definition 3. (ISS Lyapunov function) $V(\cdot)$ is an ISS Lyapunov func-

tion in the robust positive invariant set $\mathcal{O}$ for the difference inclusion $y^+ \in g(y, e)$ if there exists $\delta_e > 0$, class $\mathcal{K}_\infty$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot)$, and class $\mathcal{K}$ function $\sigma_e(\cdot)$ such that for all $y \in \mathcal{O}$ and $||\mathbf{e}|| \leq \delta_e$ which satisfy Eqs. (4.44) and (4.45).

$$\alpha_1(|y - \bar{y}|) \leq V(y) \leq \alpha_2(|y - \bar{y}|) \tag{4.44}$$

$$\sup_{y^+ \in g(y,e)} V(y^+) \leq V(y) - \alpha_3(|y - \bar{y}|) + \sigma_e(||\mathbf{e}||). \tag{4.45}$$

where $\bar{y}$ is the equilibrium point ([71]).

Proposition 3. If there exist an ISS Lyapunov function for the perturbed system $y^+ = g(y, e)$ for all $||\mathbf{e}|| \leq \delta$ for some $\delta > 0$ on a robust positive invariant set $\mathcal{O}$, then the origin of the system is robustly asymptotically stable in $\mathcal{O}$ for all $||\mathbf{e}|| \leq \delta$ ([68]).

4.3.3.1 Specification of equilibrium point and Lyapunov function candidate

We examine the closed-loop behavior of the combined system under the assumption:

Assumption 1. The approximated model-plant mismatch function $\hat{f}_d$ in Eq. (4.25) for each set-point $\bar{r}$ is completely learned.

Let $\hat{d}^{\ell^*}(\bar{r})$ denotes the disturbance value from this completely learned $\hat{f}_d$ which achieves offset-free tracking of set-point $\bar{r}$ at steady-state. Then, we can think of the target state and input $\bar{x}^{\ell^*}$ and $\bar{u}^{\ell^*}$

derived by substituting $\hat{d}^{\ell^*}$ into the target calculator in Eq. (4.7):

$$\begin{bmatrix} A-I & B \\ HC & 0 \end{bmatrix} \begin{bmatrix} \bar{x}^{\ell^*} \\ \bar{u}^{\ell^*} \end{bmatrix} = \begin{bmatrix} -B_d\hat{d}^{\ell^*} \\ \bar{r} - HC_d\hat{d}^{\ell^*} \end{bmatrix}. \tag{4.46}$$

Target $\bar{x}^{\ell^*}$ and $\bar{u}^{\ell^*}$ are the state and input pair which achieves offset-free tracking of set-point $\bar{r}$ at steady-state. Let $\bar{x}_d^{\ell^*}$ denote the augmented state of at this steady-state:

$$\bar{x}_d^{\ell^*} := \begin{bmatrix} \bar{x}^{\ell^*} \\ \hat{d}^{\ell^*} \end{bmatrix}. \tag{4.47}$$

$\bar{x}_d^{\ell^*}$ is the equilibrium point of the proposed model-plant mismatch learning offset-free MPC. Therefore, we can analyze the closed-loop stability of the proposed scheme by examining the stability of the point $\bar{x}_d^{\ell^*}$.

Then, we can consider the ideal optimal control problem P_{ℓ^*} with the ideally fixed target pair $(\bar{x}^{\ell^*}, \bar{u}^{\ell^*})$ excluding target calculator in Eq. (4.7).

$$\begin{aligned} \mathrm{P}_{\ell^*}: \ J_{\ell^*}^0(\hat{x}_d) = \min_{u_0,\cdots,u_{N-1}} \ & \phi_t^{\ell^*}(x_N) + \sum_{i=0}^{N-1} \phi^{\ell^*}(x_i, u_i) \\ \text{s.t.} \ \ & x_0 = \hat{x}, \ d = \hat{d} \\ & x_{i+1} = Ax_i + Bu_i + B_d d \\ & u_i \in \mathcal{U}, \ x_{i+1} \in \mathcal{X}, \ x_N \in \mathcal{X}_t \\ & i = 0, \ldots, N \end{aligned}$$

where $\phi^{\ell^*}(x_i, u_i) := ||x_i - \bar{x}^{\ell^*}||_{Q_x}^2 + ||u_i - \bar{u}^{\ell^*}||_{Q_u}^2$ and $\phi_t^{\ell^*}(x_N) :=$

$||x_N - \bar{x}^{\ell^*}||^2_{Q_x^N}$. The terminal constraint set $\mathcal{X}_t$ is chosen as a sublevel set of the terminal cost $\phi_t^{\ell^*}$:

$$\mathcal{X}_t = \{x \in \mathcal{X} \mid \phi_t^{\ell^*}(x) \leq \rho_t\} \tag{4.48}$$

for some $\rho_t > 0$. We specify the value function $J^0_{\ell^*}$ of P_{ℓ^*} as a Lyapunov function candidate.

4.3.3.2 Perturbation in combined system

We define the perturbation in the combined system of model-plant mismatch learning offset-free MPC for further robustness analysis. Let e_y denote the measurement error due to the intrinsic model-plant mismatch:

$$e_y := y_p - \hat{y}^{\ell,s}. \tag{4.49}$$

Let $e_{\hat{x}_d}$ denote the estimate error between real augmented state $x_d := [x^\top, d^\top]^\top$ which achieves the current plant measurement y_p and estimated augmented state $\hat{x}_d^{\ell,s} := [\hat{x}^{\ell,s\top}, \hat{d}^{\ell,s\top}]^\top$:

$$e_{\hat{x}_d} := \begin{bmatrix} x \\ d \end{bmatrix} - \begin{bmatrix} \hat{x}^{\ell,s} \\ \hat{d}^{\ell,s} \end{bmatrix}. \tag{4.50}$$

Since we can consider the model-plant mismatch as a kind of process disturbance, we assume the estimate error bound.

Assumption 2. There exists the measurement error bound δ_y such

that for all $||e_y|| \leq \delta_y$ and $k \leq 0$, the Eq. (4.51) holds.

$$|e_{\hat{x}_d}(k)| \leq \beta(|e_{\hat{x}_e}(0)|, k) + \sigma_y(||e_y||) \tag{4.51}$$

where β is a class $\mathcal{KL}$ function and σ is a class $\mathcal{K}$ function.

Not as the combined estimator/optimizer system in [68], the combined system of proposed model-plant mismatch learning offset-free MPC additionally has the target calculator. Therefore, we define and examine the influence of target error $\delta_{\bar{x},\bar{u}}$ from the ideal target pair $(\bar{x}^{\ell^*}, \bar{u}^{\ell^*})$ as in Eq. (4.52):

$$e_{\bar{x},\bar{u}} := \begin{bmatrix} \bar{x}^{\ell^*} \\ \bar{u}^{\ell^*} \end{bmatrix} - \begin{bmatrix} \bar{x}^{\ell,s} \\ \bar{u}^{\ell,s} \end{bmatrix}. \tag{4.52}$$

Remark 5. Substituting Eq. (4.26) into the right-hand side of Eq. (4.46) and rearranging yields

$$\begin{bmatrix} -B_d\hat{d}^{\ell,s} \\ \bar{r} - HC_d\hat{d}^{\ell,s} \end{bmatrix} = \begin{bmatrix} -B_d\hat{d}^{\ell^*} \\ \bar{r} - HC_d\hat{d}^{\ell^*} \end{bmatrix} + \begin{bmatrix} -B_d\hat{d}^{s} \\ -HC_d\hat{d}^{s} \end{bmatrix}. \tag{4.53}$$

Then, we can derive Eq. (4.54) by substituting Eqs. (4.7), (4.52) and (4.53) into Eq. (4.46) and rearranging.

$$\begin{bmatrix} A - I & B \\ HC & 0 \end{bmatrix} e_{\bar{x},\bar{u}} = \begin{bmatrix} B_d \\ HC_d \end{bmatrix} \hat{d}^s. \tag{4.54}$$

We can see $e_{\bar{x},\bar{u}}$ is mainly dependent on $\hat{d}^s$.

By **Remark 5**, we examine $e_{\bar{x},\bar{u}}$ by analyzing behavior of $\hat{d}^s$. In the proposed scheme, $\hat{d}^\ell$ acts as an warm-start signal in disturbance estimator. Therefore, since the completely learned $\hat{d}^{\ell^*}$ as in **Assumption 1** can guarantee considerable prediction accuracy, supplementary signal $\hat{d}^s$ has sufficiently small value. Therefore, we assume that $\hat{d}^s$ is bounded (We can see $\hat{d}^s$ is actually bounded near the origin in numerical examples in *Section 4.4*), and thus, we also assume that $e_{\bar{x},\bar{u}}$ is bounded.

Now, we examine the influence of $e_{\bar{x},\bar{u}}$ on the closed-loop behavior of the combined system. Let $\kappa_p^{\ell^*}$ and $\kappa_p^{\ell,s}$ denote the control laws of P_{ℓ^*} and P_0, respectively. Then, we can define the error of control law as:

$$e_{\kappa_p} := \kappa_p^{\ell,s} - \kappa_p^{\ell^*}. \tag{4.55}$$

Since we assume that $e_{\bar{x},\bar{u}}$ is bounded, we can also assume that e_{κ_p} is bounded:

Assumption 3. There exists $\hat{d}^s$ bound $\delta_{\hat{d}^s}$ such that for all $||\hat{d}^s|| \leq \delta_{\hat{d}^s}$ and $k \leq 0$, the Eq. (4.56) holds.

$$|e_{\kappa_p}(k)| \leq \sigma_{\hat{d}^s}(||\hat{d}^s||) \tag{4.56}$$

where σ is a class $\mathcal{K}$ function.

In **Assumption 3**, we consider the $\hat{d}^s$ as a kind of error source, since $e_{\bar{x},\bar{u}}$ mainly depends on $\hat{d}^s$ as in **Remark 5**.

Then, we can denote the estimate of evolved state $\hat{x}_d^+$ through the

proposed combined system with perturbation in estimate, target, and measurement:

$$\hat{x}_d^+ := f(\hat{x}_d + e_{\hat{x}_d}, \kappa_p^{\ell^*}(\hat{x}_d) + e_{\kappa_p}) + e_y^L - e_{\hat{x}_d}^+ \tag{4.57}$$

$$f(x_d, u) := \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} x_d + \begin{bmatrix} B \\ 0 \end{bmatrix} u. \tag{4.58}$$

where $e_y^L := \begin{bmatrix} L_x \\ L_d \end{bmatrix} e_y$.

4.3.3.3 Robust asymptotic stability of model-plant mismatch learning offset-free MPC

In this section, we show the robust asymptotic stability of the equilibrium point $\bar{x}_d^{\ell^*}$ by establishing that $J_{\ell^*}^0$ is an ISS Lyapunov function for the proposed combined system.

We can easily show that $J_{\ell^*}^0$ satisfies the upper and lower bounding inequality in Eq. (4.44) with **Proposition 4**:

$$\alpha_1(|x_d - \bar{x}_d^{\ell^*}|) \leq J_{\ell^*}^0(x_d) \leq \alpha_2(|x_d - \bar{x}_d^{\ell^*}|) \tag{4.59}$$

Proposition 4. Suppose $V : \mathbb{R}^n \to \mathbb{R}$ to be a continuous positive definite function defined on $\mathbb{R}^n$ and radially unbounded. Then, there exist class $\mathcal{K}_\infty$ functions α_1 and α_2 which satisfy the lower and upper bounding inequality in Eq. (4.60).

$$\alpha_1(|y - \bar{y}|) \leq V(y) \leq \alpha_2(|y - \bar{y}|) \tag{4.60}$$

where $\bar{y}$ is the equilibrium point. The proof is provided in *Appendix C.4* of [72].

Now, let $\tilde{x}_d^+$ denote the expected evolved state with the control law $\kappa_p^{\ell^*}$ of the ideal problem P_{ℓ^*}:

$$\tilde{x}_d^+ := f(\hat{x}_d, \kappa_p^{\ell^*}(\hat{x}_d)). \tag{4.61}$$

We first suggest a standard feasible solution $\tilde{\mathbf{u}}^+$ for $\tilde{x}_d^+$ and show that $\tilde{\mathbf{u}}^+$ is robustly feasible for $\hat{x}_d^+$. And then, we prove that $J_0^{\ell^*}(\hat{x}_d^+)$ satisfies the inequality in Eq. (4.45) using $\tilde{\mathbf{u}}^+$ under the assumptions:

Assumption 4. The model-plant mismatch is not severe, so that $\hat{d}^{\ell,s}$ is always in a compact set $\mathcal{D} \in \mathbb{R}^{n_d}$.

Assumption 5. Let $\mathcal{X}_t^D := \mathcal{X}_t \times \mathcal{D}$, $\phi_d^{\ell^*}(x_d) := \phi^{\ell^*}(Sx_d)$, and $\phi_{d,t}^{\ell^*}(x_d) := \phi_t^{\ell^*}(Sx_d)$ where $S := [I_{n_x}, \mathbf{0}_{n_x \times n_d}]$. For all $x_d \in \mathcal{X}_t^D$, there exist a local control law $\kappa_t : \mathcal{X}_t^D \to \mathcal{U}$ satisfying

$$f(x_d, \kappa_t(x_d)) \in \mathcal{X}_t^D \tag{4.62}$$

$$\phi_{d,t}^{\ell^*}(f(x_d, \kappa_t(x_d))) \leq \phi_{d,t}^{\ell^*}(x_d) - \phi_d^{\ell^*}(x_d, \kappa_t(x_d)). \tag{4.63}$$

Assumption 6. Let $\mathcal{X}^D := \mathcal{X} \times \mathcal{D}$. There exists a $\mathcal{K}_\infty$ function α_ϕ satisfying the inequality in Eq. (4.64) for all $x_d \in \mathcal{X}^D$ and $u \in \mathcal{U}$.

$$\phi_d^{\ell^*}(x_d, u) \geq \alpha_\phi(|x_d - \bar{x}_d^{\ell^*}|). \tag{4.64}$$

Proposition 5. Let $\mathcal{Y}$ compact metric space and $g : \mathcal{Y} \to \mathbb{R}^n$ continuous. Then, there exists a class $\mathcal{K}$ function σ such that $|g(p) - g(q)| \leq \sigma(|p - q|)$ for all $p, q \in \mathcal{Y}$.

Proof). Since g is uniformly continuous on $\mathcal{Y}$ by *Theorem 4.19* in [73], for every ε there exists $\delta > 0$ such that $|g(p) - g(q)| < \varepsilon$ for all $p, q \in \mathcal{Y}$ for which $|p - q| < \delta$. Then, we can see there exists a local overbounding class $\mathcal{K}$ function $\bar{\sigma}$ and $\bar{\delta} > 0$ such that $|g(p) - g(q)| \leq \bar{\sigma}(|p - q|)$ for all $p, q \in \mathcal{Y}$ for which $|p - q| < \bar{\delta}$ by *Proposition 5* in [74]. Finally, as the similar manner in the third part of proof in *Proposition 20* in [68], we can find the global overbounding class $\mathcal{K}$ function σ such that $|g(p) - g(q)| \leq \sigma(|p - q|)$ for all $p, q \in \mathcal{Y}$. □

Proposition 6. Let e_c denotes the perturbations in state transition for the combined disturbance estimator/target calculator/optimizer system:

$$e_c := (e_{\hat{x}_d}, e_{\kappa_p}, e_y^L, e_{\hat{x}_d}^+). \tag{4.65}$$

Then, for $\tilde{x}_d^+$ and $\hat{x}_d^+$, there exists a class $\mathcal{K}$ function σ_{x^+} satisfying the inequality:

$$|\hat{x}_d^+ - \tilde{x}_d^+| \leq \sigma_{x^+}(|e_c|). \tag{4.66}$$

Proof). From Eqs. (4.57) and (4.61), we have

$$\begin{aligned}|\hat{x}_d^+ - \tilde{x}_d^+| &= |f(\hat{x}_d + e_{\hat{x}_d}, \kappa_p^{\ell^*}(\hat{x}_d) + e_{\kappa_p}) + e_y^L - e_{\hat{x}_d}^+ \\ &\quad - f(\hat{x}_d, \kappa_p^{\ell^*}(\hat{x}_d))| \\ &\leq |f(\hat{x}_d + e_{\hat{x}_d}, \kappa_p^{\ell^*}(\hat{x}_d) + e_{\kappa_p}) - f(\hat{x}_d, \kappa_p^{\ell^*}(\hat{x}_d))| \\ &\quad + |e_y^L| + |e_{\hat{x}_d}^+| \end{aligned} \tag{4.67}$$

Since $\mathcal{X}^D$ and $\mathcal{U}$ are compact sets in metric space and $f(x_d, u)$ is continuous, from **Proposition 5**, there exists a class $\mathcal{K}$ function such that

$$\begin{aligned}|f(\hat{x}_d + e_{\hat{x}_d}, \kappa_p^{\ell^*}(\hat{x}_d) + e_{\kappa_p}) - f(\hat{x}_d, \kappa_p^{\ell^*}(\hat{x}_d))| \\ \leq \sigma_f(|(e_{\hat{x}_d}, e_{\kappa_p})|).\end{aligned} \tag{4.68}$$

By substituting Eq. (4.68) into Eq. (4.67) and applying Eq. (4.65), we can derive Eq. (4.69).

$$\begin{aligned}|\hat{x}_d^+ - \tilde{x}_d^+| &\leq \sigma_f(|(e_{\hat{x}_d}, e_{\kappa_p})|) + |e_y^L| + |e_{\hat{x}_d}^+| \\ &\leq \sigma_f(|e_c|) + |e_c| + |e_c| \\ &\leq \sigma_{x^+}(|e_c|) \end{aligned} \tag{4.69}$$

where $\sigma_{x^+}(z) := \sigma_f(z) + 2z$. □

Lemma 2. Let $\mathbf{u}^0 := [u_0^{0\top}, u_1^{0\top}, \cdots, u_{N-1}^{0\top}]^\top$ denote the optimal solution sequence from P_{ℓ^*} for $\hat{x}_d$, and $\mathbf{x}_d^0 := [\hat{x}_{d,1}^{0\top}, \hat{x}_{d,2}^{0\top}, \cdots, \hat{x}_{d,N}^{0\top}]^\top$ denote the resultant augmented state sequence where $\hat{x}_{d,k}^0 := \eta(k, \hat{x}_d, \mathbf{u}^0)$ and $\eta(k, x_d, \mathbf{u})$ denotes the open-loop solution of Eq. (4.58) for given

step k from x_d with input sequence $\mathbf{u}$. We suppose a standard feasible solution for $\tilde{x}_d^+$ with a local control law κ_t in **Assumption 4**:

$$\tilde{\mathbf{u}}^+ := [u_1^{0\top}, u_2^{0\top}, \cdots, u_{N-1}^{0\top}, \kappa_t(\hat{x}_{d,N}^0)^\top]^\top. \tag{4.70}$$

Then, $\tilde{\mathbf{u}}^+$ is robustly feasible for $\hat{x}_d^+$ when the perturbation $|e_c|$ is sufficiently small.

Proof). Since $\hat{x}_{d,N}^0 \in \mathcal{X}_t^D$, Eq. (4.71) holds from Eqs. (4.63) and (4.64) in **Assumption 5, 6**.

$$\phi_{d,t}^{\ell^*}(f(\hat{x}_{d,N}^0, \kappa_t(\hat{x}_{d,N}^0))) \leq \phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^0) - \alpha_\phi(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) \tag{4.71}$$

Let $\hat{x}_{d,k}^+ := \eta(k, \hat{x}_d^+, \tilde{\mathbf{u}}^+)$, and $\tilde{x}_{d,k}^+ := \eta(k, \tilde{x}_d^+, \tilde{\mathbf{u}}^+)$. Since $\tilde{x}_{d,N-1}^+ = \hat{x}_{d,N}^0$ and $\tilde{x}_{d,N}^+ = f(\hat{x}_{d,N}^0, \kappa_t(\hat{x}_{d,N}^0))$ from Eq. (4.61), Eq. (4.72) holds.

$$\phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+) \leq \phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^0) - \alpha_\phi(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) \tag{4.72}$$

From $\phi_{d,t}^{\ell^*}(\bar{x}_d^{\ell^*}) = 0$ and **Proposition 5**, there exists a class $\mathcal{K}_\infty$ function α_t such that

$$\phi_{d,t}^{\ell^*}(x_d) \leq \alpha_t(|x_d - \bar{x}_d^{\ell^*}|) \tag{4.73}$$

for all $x_d \in \mathcal{X}_t^D$. Substituting Eq. (4.73) into Eq. (4.72) yields:

$$\begin{aligned} &\phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+) \leq \alpha_{t,\phi}(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) \\ &\alpha_{t,\phi}(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) := \alpha_t(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) - \alpha_\phi(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) \end{aligned} \tag{4.74}$$

Now, from **Proposition 5**, there exists a class $\mathcal{K}$ function σ_t such that

$$|\phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^+) - \phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+)| \leq \sigma_t(|\hat{x}_{d,N}^+ - \tilde{x}_{d,N}^+|). \tag{4.75}$$

Since $\sigma_t \geq 0$, Eq. (4.76) holds in both cases where $\phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^+) - \phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+)$ is positive or negative.

$$\phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^+) \leq \phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+) + \sigma_t(|\hat{x}_{d,N}^+ - \tilde{x}_{d,N}^+|). \tag{4.76}$$

Substituting Eq. (4.69) in **Proposition 6** into Eq. (4.76) yields:

$$\begin{aligned} &\phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^+) \leq \phi_{d,t}^{\ell^*}(\tilde{x}_{d,N}^+) + \sigma_{t,x^+}(|e_c|) \\ &\sigma_{t,x^+}(|e_c|) := \sigma_t(\sigma_{x^+}(|e_c|)). \end{aligned} \tag{4.77}$$

Then, substituting Eq. (4.74) into Eq. (4.77) yields:

$$\phi_{d,t}^{\ell^*}(\hat{x}_{d,N}^+) \leq \alpha_{t,\phi}(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|) + \sigma_{t,x^+}(|e_c|). \tag{4.78}$$

Therefore, since we set $\mathcal{X}_t^D = \{x_d|\ \phi_t^{\ell^*}(x_d) \leq \rho_t\}$, we can see if Eq. (4.79) holds, then $\hat{x}_{d,N}^0 \in \mathcal{X}_t^D$ implies $\hat{x}_{d,N}^+ \in \mathcal{X}_t^D$.

$$|e_c| \leq \sigma_{t,x^+}^{-1}(\rho_t - \alpha_{t,\phi}(|\hat{x}_{d,N}^0 - \bar{x}_d^{\ell^*}|)). \tag{4.79}$$

□

Lemma 3. Suppose $\hat{x}_d \in \mathcal{X}^D := \{x_d|\ J_{\ell^*}^0(x_d) \leq \rho_J\}$. Then, $\mathcal{X}^D$ is robustly positive invariant and $J_0^{\ell^*}$ satisfies Eq. (4.80) with a class

$\mathcal{K}_\infty$ function α and a class $\mathcal{K}$ function σ.

$$J^0_{P_{\ell^*}}(\hat{x}^+_d) \leq J^0_{P_{\ell^*}}(\hat{x}_d) - \alpha(|\hat{x}_d|) + \sigma(||\mathbf{e}_c||) \tag{4.80}$$

where $\mathbf{e}_c$ is the sequence of combined error e_c in Eq. (4.65).

Proof). Let $x_{d,k} := \eta(k, x_d, \mathbf{u})$ and $J_{\ell^*}(x_d, \mathbf{u}) := \phi^{\ell^*}_{d,t}(x_{d,N}) + \sum_{i=0}^{N-1} \phi^{\ell^*}_d(x_{d,i}, u_i)$ denote the resultant cost sum from x_d with solution sequence $\mathbf{u}$. Then, we can derive Eq. (4.81).

$$\begin{aligned} &J_{\ell^*}(\tilde{x}^+_d, \mathbf{u}^+) - J^0_{\ell^*}(\hat{x}_d) + \phi^{\ell^*}_d(\hat{x}_d, \kappa^{\ell^*}_p(\hat{x}_d)) \\ &= \phi^{\ell^*}_d(\hat{x}^0_{d,N}, \kappa_t(\hat{x}^0_{d,N})) + \phi^{\ell^*}_{d,t}(\tilde{x}^+_{d,N}) - \phi^{\ell^*}_{d,t}(\hat{x}^0_{d,N}) \end{aligned} \tag{4.81}$$

Substituting Eq. (4.81) into Eq. (4.63) and rearranging yields:

$$J_{\ell^*}(\tilde{x}^+_d, \mathbf{u}^+) \leq J^0_{\ell^*}(\hat{x}_d) - \phi^{\ell^*}_d(\hat{x}_d, \kappa^{\ell^*}_p(\hat{x}_d)) \tag{4.82}$$

Then, by substituting Eq. (4.64) into Eq. (4.82), we can derive:

$$J_{\ell^*}(\tilde{x}^+_d, \mathbf{u}^+) \leq J^0_{\ell^*}(\hat{x}_d) - \alpha_\phi(|\hat{x}_d - \bar{x}^{\ell^*}_d|) \tag{4.83}$$

Now, since $J_{\ell^*}(x_d, u)$ is continuous, there exists a class $\mathcal{K}$ function σ_J from **Proposition 5**.

$$|J_{\ell^*}(\hat{x}^+_d, \mathbf{u}^+) - J_{\ell^*}(\tilde{x}^+_d, \mathbf{u}^+)| \leq \sigma_J(|\hat{x}^+_d - \tilde{x}^+_d|). \tag{4.84}$$

We can drop the absolute value as similar manner in the proof of

Lemma 2:

$$J_{\ell^*}(\hat{x}_d^+, \mathbf{u}^+) \leq J_{\ell^*}(\tilde{x}_d^+, \mathbf{u}^+) + \sigma_J(|\hat{x}_d^+ - \tilde{x}_d^+|). \tag{4.85}$$

Since $J_{\ell^*}(\hat{x}_d^+, \mathbf{u}^+) \geq J_{\ell^*}^0(\hat{x}_d^+)$ and $|e_c| \leq ||\mathbf{e}_c||$, substituting Eqs. (4.49) and (4.83) into Eq. (4.85) and rearranging yields:

$$J_{\ell^*}^0(\hat{x}_d^+) \leq J_{\ell^*}^0(\hat{x}_d) - \alpha_\phi(|\hat{x}_d - \bar{x}_d^{\ell^*}|) + \sigma_{J,x^+}(||\mathbf{e}_c||). \tag{4.86}$$

where $\sigma_{J,x^+}(|e_c|) := \sigma_J(\sigma_{x^+}(|e_c|))$ is a class $\mathcal{K}$ function. Therefore, we can see that $J_{\ell^*}^0(\cdot)$ satisfies the Eq. (4.45).

Additionally, we can see if Eq. (4.87) holds, then $\hat{x}_d \in \mathcal{X}^D$ implies $\hat{x}_d^+ \in \mathcal{X}^D$.

$$|e_c| \leq \sigma_{J,x^+}^{-1}(\rho_J - J_{\ell^*}^0(\hat{x}_d) + \alpha_\phi(|\hat{x}_d - \bar{x}_d^{\ell^*}|)). \tag{4.87}$$

□

Theorem 4. We have established that the value function of the ideal optimal control problem $J_{\ell^*}^0$ from the perfectly learned steady-state model-plant mismatch is an ISS Lyapunov function in the robust invariant set $\mathcal{X}^D$ of the combined disturbance estimator/target calculator/optimizer system of the proposed model-plant mismatch learning offset-free MPC through **Lemma 2, 3**. Finally, by **Proposition 3**, we can see the equilibrium point $\bar{x}_d^{\ell^*}$ of the combined system is robustly asymptotically stable.

4.4 Numerical example

In this section, we demonstrate three numerical simulation results to demonstrate the performance of the proposed model-plant mismatch learning offset-free MPC. The first case shows the model-plant mismatch learning and tracking performance for a randomly changing set-point, the second case shows the learning efficacy with data window strategy when the plant deformation occurs during operation, and the third case shows the performance with two randomly changing set-points.

We consider a continuous stirred-tank reactor (CSTR) where the first-order reaction, $A \rightarrow B$ takes place in the liquid phase, and the reactor temperature is controlled with the external cooling jacket in [69].

The control objective is to track the reference trajectories of the outlet concentration of reactant (c) while regulating the reactor temperature (T) as a fixed value by directly manipulating the temperature of the jacket (T_c) and the outlet flow rate (F). The following equations describe the dynamics of the reactor.

$$\begin{cases} \dfrac{dc}{dt} = \dfrac{F_0(c_0 - c)}{\pi r^2 h} - k_0 \exp(-\dfrac{E}{RT})c \\ \dfrac{dT}{dt} = \dfrac{F_0(T_0 - T)}{\pi r^2 h} - \dfrac{\Delta H}{\rho C_p} k_0 \exp(-\dfrac{E}{RT})c \\ \qquad\quad + \dfrac{2U}{r\rho C_p}(T_c - T) \\ \dfrac{dh}{dt} = \dfrac{F_0 - F}{\pi r^2} \end{cases} \tag{4.88}$$

The parameters are given in **Table 4.1**.

Parameter	Nominal value	Units
F_0	0.1	$\mathrm{m}^3/\mathrm{min}$
T_0	350	K
c_0	1	$\mathrm{kmol}/\mathrm{m}^3$
r	0.219	m
k_0	7.2×10^{10}	min^{-1}
E/R	8750	K
U	54.94	$\mathrm{kJ}/\mathrm{min}\cdot\mathrm{m}^2\cdot\mathrm{K}$
ρ	1000	kg/m^3
C_p	0.239	$\mathrm{kJ}/\mathrm{kg}\cdot\mathrm{K}$
ΔH	-5×10^4	$\mathrm{kJ}/\mathrm{kmol}$

Table 4.1: Parameters of the CSTR.

The linear model is derived at the steady-state:

$$c^s = 0.878\ \mathrm{kmol}/\mathrm{m}^3, \quad T^s = 324.5\ \mathrm{K}, \quad h^s = 0.659\ \mathrm{m}$$
$$T_c^s = 300\ \mathrm{K}, \quad F^s = 0.1\ \mathrm{m}^3/\mathrm{min}$$

The discretized linear model in Eq. (4.89) with sampling instant 1min is used for MPC.

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \\ z(k) = Hy(k) \end{cases} \tag{4.89}$$

$$A = \begin{bmatrix} 0.2681 & -0.00338 & -0.00728 \\ 9.703 & 0.3279 & -25.44 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.00537 & 0.1655 \\ 1.297 & 97.91 \\ 0 & -6.637 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The following operational constraints are applied to the system:

$$0.83 \leq c \leq 0.92,\ 320 \leq T \leq 330,\ 0.4 \leq h \leq 1.2,$$
$$295 \leq T_c \leq 310,\ 0.07 \leq F \leq 0.13$$

The prediction horizon and weights in optimal control problem are $N = 10$, $Q_x = diag\{50; 0.001; 1\}$, $Q_x^N = 10Q_x\}$ and $Q_u = diag\{0.01; 0.01\}$, respectively.

For disturbance estimator, following disturbance model and gain

matrices are applied:

$$B_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \; C_d = \mathbf{0}_{n_y \times n_d},$$

$$L_x = \begin{bmatrix} 0.6141 & -0.0034 & -0.0071 \\ 9.7056 & 0.7346 & -25.4413 \\ 0 & 0 & 0.2800 \end{bmatrix},$$

$$L_d = \begin{bmatrix} 0.3420 & 0 & 0 \\ 0 & 0.4026 & 0 \end{bmatrix}$$

We also suppose that the outlet concentration is 3% higher than the average concentration in reactor, $F \rightarrow 1.03\,F$, to apply additional error and increase the model-plant mismatch.

Simulations are performed using MATLAB® R2019a with Intel® Core™ i7-6700 CPU @ 3.40GHz, 32 GB RAM.

4.4.1 System with random set-point

We applied the random set-point of c along $[0.84, 0.91]$ with fixed set-point of T for every 15 sampling instance, and approximate intrinsic model-plant mismatch f_d in Eq. (4.22) using GRNN from the estimated disturbance data at steady-state for each c set-point.

Figure 4.1 illustrates the results of function approximation for f_d from 10 and 50 steady-state disturbance data, respectively, for each random c set-point, c_{ref}. We can see the estimated disturbance data and set-point shows a certain relation which indicates the steady-state model-plant mismatch for each set-point between Eqs. (4.88)

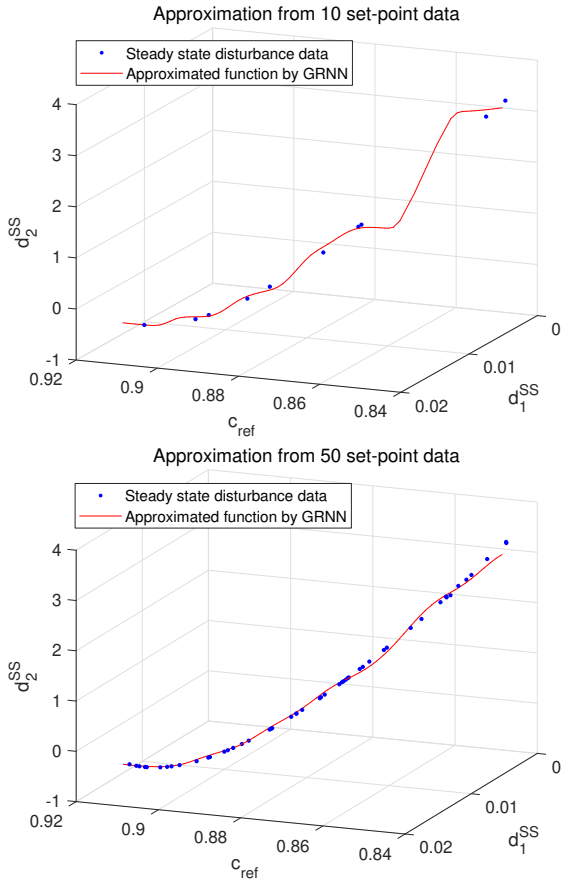

Figure 4.1: Approximated model-plant mismatch by GRNN from 10 and 50 disturbance data at steady-state for each random set-point.

and (4.89). The approximated function by GRNN follows the trend of the data well with sufficient data.

Figure 4.2 shows the reference tracking results of the nominal offset-free MPC and the proposed model-plant mismatch learning offset-free MPC which applies learned disturbance value for each set-point c approximated by GRNN with 50 steady-state data in **Figure 4.1**. Though all the schemes achieve offset-free tracking, the nominal scheme arouses considerable error when the set-point changes because the nominal method gradually estimates the model-plant mismatch for each set-point only from occurred measurement error. On the other hand, the proposed scheme tracks the reference trajectory with only tiny error at the transition state and shows much better performance compared to the standard method. The reason is that the proposed scheme learns the intrinsic model-plant mismatch from data and applies the learned information into the estimator and finite-horizon optimal control problem.

From **Figure 4.3** which describes the estimated disturbance values of each scheme, we can examine how the proposed scheme improved the controller performance in more detail. 'Nominal D' and 'Combined D' imply the estimated disturbance from the nominal offset-free MPC and the proposed model-plant mismatch learning offset-free MPC, respectively. We can see the proposed scheme shows considerably superior performance in disturbance estimation. 'Combined D' is the sum of learned steady-state disturbance from GRNN 'Learned D' and supplementary disturbance estimated from the estimator in Eq. (4.27) 'Supple. D'. We can see the learned disturbance acts as a warm-start signal so that the supplementary disturbance only changes in much smaller range than the fully estimated disturbance

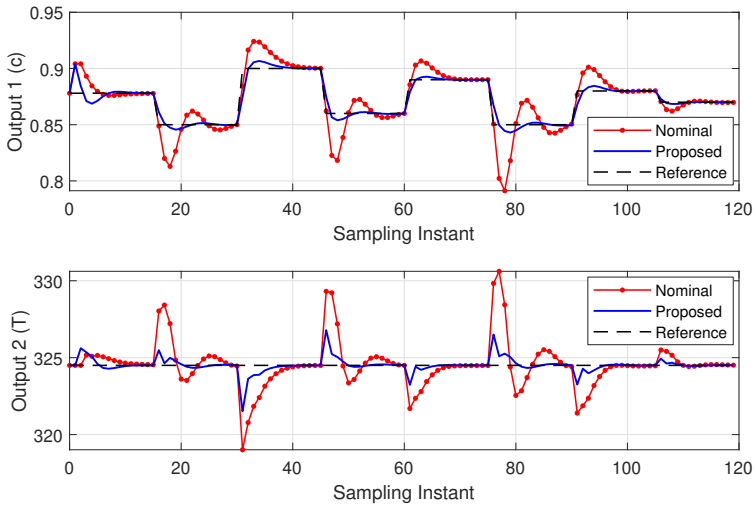

Figure 4.2: Reference tracking results of offset-free MPC for the random set-point of c and fixed T.

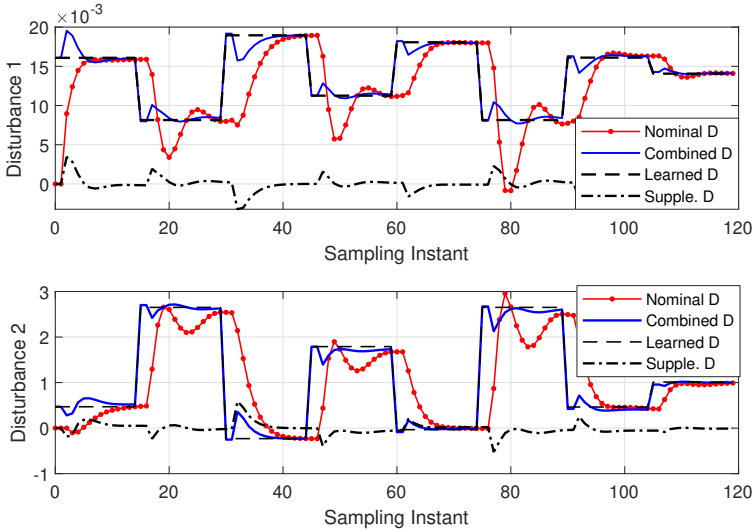

Figure 4.3: Estimated disturbance of offset-free MPC schemes.

of the nominal scheme. By this, the proposed scheme effectively improves the model-plant mismatch compensating performance of the disturbance estimator and prediction accuracy in finite-horizon optimal control problem.

4.4.2 Transformed system

In this section, we examine the model-plant mismatch learning performance with data-window strategy and reference tracking performance when system transformation occurs. We changed the reaction rate constant k_0 in **Table 1** from $7.2 \times 10^{10}\mathrm{min}^{-1}$ to $6.2 \times 10^{10}\mathrm{min}^{-1}$ to implement the system transformation into the plant. Then, we applied the random set-point of c along $[0.84, 0.91]$ with fixed set-point of T for every 15 sampling instance as similar in the previous section, and approximate the intrinsic model-plant mismatch f_d using GRNN for each c set-point with data-window strategy.

Figure 4.4 illustrates the result of approximation for the steady-state model-plant mismatch for each set-point of c applying data-window strategy which replaces the old data into the recent data. We can see the approximated function $\hat{f}_d$ gradually transforms from that of original system to that of transformed system as the sampled data from the original system is replaced into the data from the transformed system.

Figure 4.5 describes the reference tracking result of nominal and proposed offset-free MPC schemes where 'Learned ori.' and 'Learned trs.' imply the closed-loop results with learned model-plant mismatch from the 50 original system data and the 50 transformed system data as in **Figure 4.4**, respectively. The proposed scheme with learned dis-

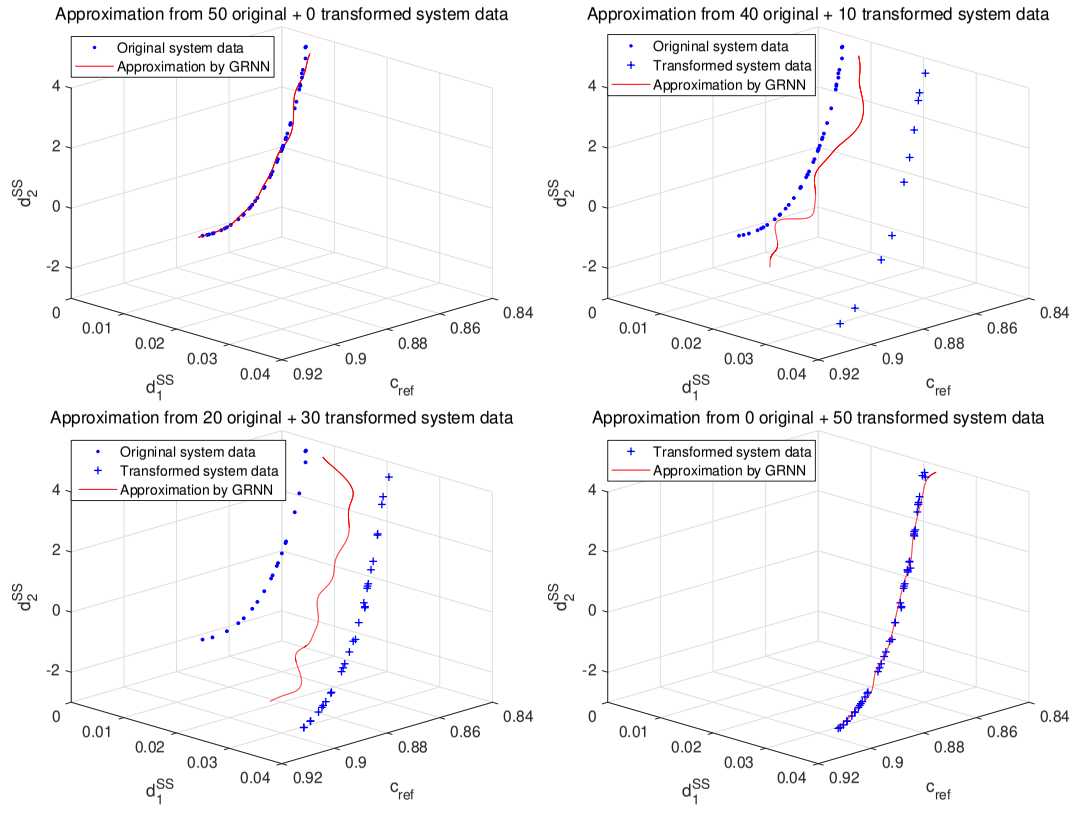

Figure 4.4: Approximated model-plant mismatch transformation from the original system to the transformed system.

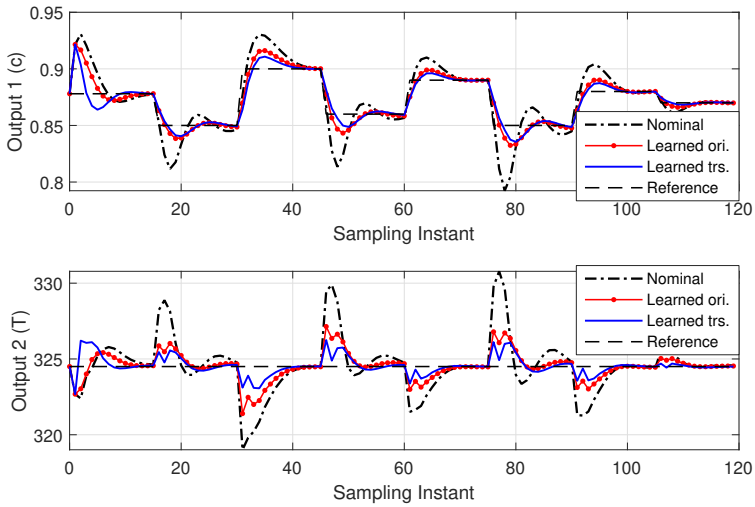

Figure 4.5: Reference tracking results of offset-free MPC for the random set-point of c and fixed T in the transformed system.

turbance from the transformed system data shows superior performance than that from the original system data. Therefore, we can see the proposed scheme with data-window strategy works fine under the system transformation.

Figure 4.6 shows the estimated disturbance values of each scheme. We can see the learned disturbance value from the transformed the system data matches the steady-state disturbance well than that from the original system data, so that the scheme based on the transformed data can compensate the model-plant mismatch using smaller amount of supplementary disturbance than that based on the original system data. By this, the proposed scheme with data-window strategy can handle the system transformation well.

Additionally, even the scheme based on the original system data shows considerably superior performance than that of the nominal scheme in **Figure 4.5**. Though the learned disturbance signal from the original data does not match the steady-state value of the transformed system, it can work as a proper warm-start signal for the disturbance estimator. This implies that even the learned model-plant mismatch without information of system transformation can improve the closed-loop performance of the controller in the circumstance where the transformation is not that considerable.

4.4.3 System with multiple random set-points

In this section, we examine the closed-loop behavior of offset-free MPC schemes in system with multiple random set-points. We applied the random set-point of c along $[0.84, 0.91]$ and T along $[321, 329]$ for every 15 sampling instance, and approximate intrinsic model-

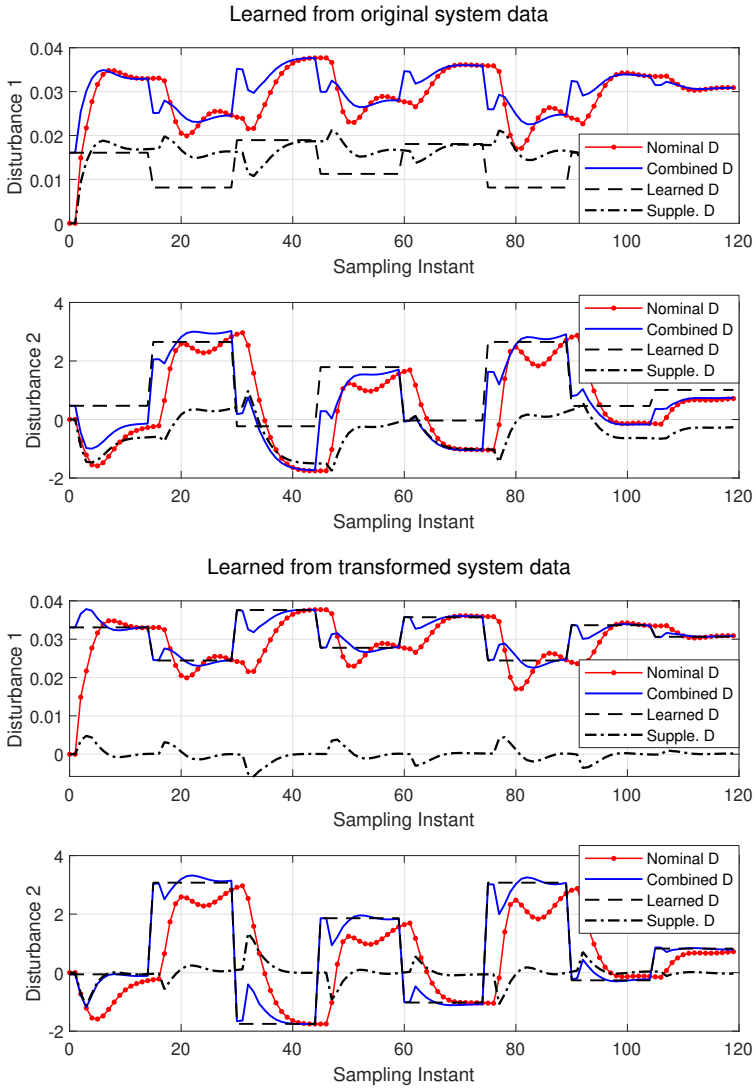

Figure 4.6: Estimated disturbance of offset-free MPC schemes for the transformed system.

plant mismatch f_d using GRNN from the estimated disturbance data at steady-state for each (c, T) set-point pair.

Figure 4.7 illustrates the approximation result of the steady-state model-plant mismatch f_d by GRNN. Since the disturbance pair (d_1^{ss},d_2^{ss}) depends on each (c_{ref},T_{ref}) pair, d_1^{ss} and d_2^{ss} for each (c_{ref},T_{ref}) pair are illustrated separately. We can see the approximated function $\hat{f}_d$ by GRNN follows the trend of the data well in the system with multiple random set-points.

Figure 4.8 shows the reference tracking results of nominal and proposed offset-free MPC schemes with learned model-plant mismatch from 100 and 400 steady-state data, respectively. The both proposed schemes applying learned model-plant mismatch from 100 and 400 data show considerably superior performance than that of the nominal scheme. Though the approximated function with 400 data is more exact and smooth than that of 100 data, the closed-loop performances of controllers with these approximated functions are almost identical. Therefore, we can see the scattered 100 data is sufficient to learn a proper approximation of model-plant mismatch.

Figure 4.9 shows the estimated disturbance values of nominal and proposed scheme with 100 steady-state disturbance data in **Figure 4.7**. The learned disturbance value matches the steady-state disturbance properly, but there are some deviations near 40^{th}, 80^{th}, and 160^{th} sampling instants. This implies that the model-plant mismatch map is not perfectly learned near the related set-points. However, the proposed scheme achieves the offset-free tracking performance and efficiently improves the closed-loop performance as in **Figure 4.8** by exploiting the supplementary disturbance values in the system with multiple random set-points.

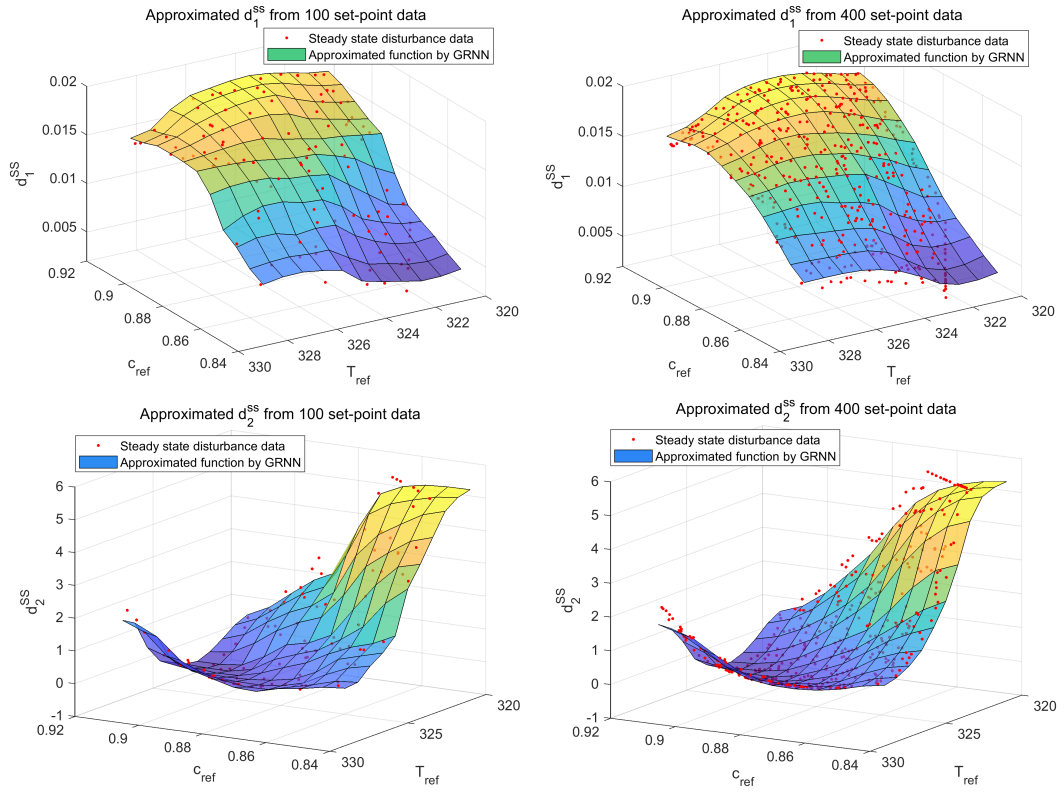

Figure 4.7: Approximated model-plant mismatch by GRNN from 100 and 400 disturbance data at steady-state for each multiple random set-points.

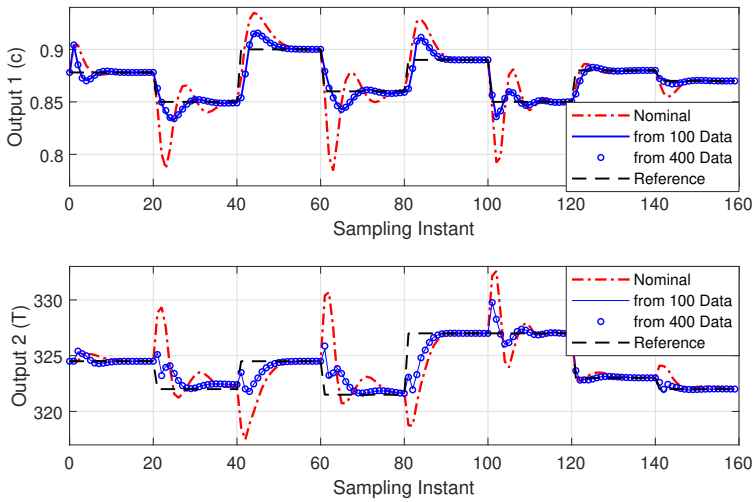

Figure 4.8: Reference tracking results of offset-free MPC for the random set-point of c and T.

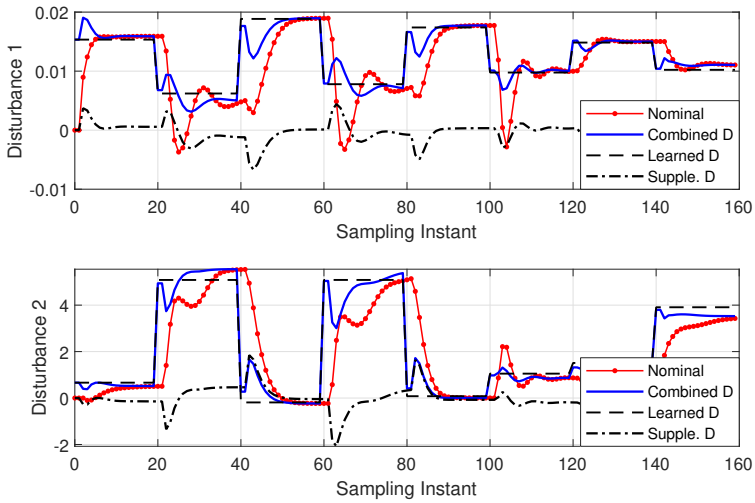

Figure 4.9: Estimated disturbance of offset-free MPC schemes for the random set-point of c and T.

Chapter 5

Concluding remarks

5.1 Move-blocked model predictive control with linear interpolation of base sequences

This study presented the analysis of existing offset blocking schemes that are commonly used in the field of constrained move blocked MPC. The offset blocking schemes make it possible to guarantee recursive feasibility and overcome the drawbacks of input blocking schemes by utilizing the valuable properties from the base sequence. However, since existing schemes are trapped in formulations that use only a fixed base sequence and do not fully exploit valuable properties from various base sequences, they have limitations in terms of optimality and could degrade the performance of the controller. Thus, we analyzed existing offset blocking schemes from the viewpoint of cost optimality and proposed the interpolated solution based offset blocking strategy to address these limitations.

The interpolated solution based offset blocking strategy parameterizes the input sequence in terms of deviations from the convex combination of the infinite-horizon LQR solution and the retained optimal solution from the previous sampling instant. The proposed interpolated solution based move blocked MPC always guarantees

recursive feasibility and stability by utilizing the feasibility of the shifted previous solution with considerably larger feasible region than the existing LQR solution based move blocked MPC, and also efficiently improves the optimality performance of the controller by utilizing the closed-loop optimality of the LQR solution compared to the existing previous solution based move blocked MPC. Moreover, the interpolated solution based move blocked MPC can further improve the optimality performance by applying the concept of dual-mode control.

The numerical examples show that the proposed interpolated solution based move blocked MPC efficiently enlarge the feasible region and improve the optimality performance compare to the existing move blocked MPC schemes. In conclusion, the interpolated solution based move blocked MPC can be a useful alternative in the field of move blocking for computationally-efficient MPC.

5.2 Move-blocked model predictive control with time-varying blocking structure by semi-explicit approach

This study proposed the semi-explicit approach for move blocked MPC to improve the optimality performance of the controller by selecting an appropriate time-varying blocking structure for the system state while moving the on-line computational complexity for deriving the suitable blocking structure to off-line by solving multiparametric program.

Since the optimal control problem of move blocked MPC is an MIP, we have to consider all the combinations of admissible blocking structure and the active constraint set to explicitly specify the

optimizer of the problem. Therefore, we proposed the semi-explicit approach to move the on-line computational cost for deriving the appropriate blocking structure to off-line. By this, we can considerably reduce the on-line computation complexity by converting the MIP to a simple convex optimization problem while avoiding to derive an excessive number of critical regions.

The numerical examples show that the proposed semi-explicit move blocked MPC achieves better closed-loop performance than the commonly used time-invariant blocking structure case. In conclusion, semi-explicit move blocked MPC can be an effective alternative to improve the optimality performance of the controller while achieving computational-efficiency. Moreover, it is expected to be more valuable in the situation where MPC is directly implemented at the actuator level with the recent development of Internet of Things technology.

5.3 Model-plant mismatch learning offset-free model predictive control

We propose a novel offset-free MPC scheme which learns the intrinsic steady-state model-plant mismatch from data and applies the learned information into the offset-free model predictive controller with supplementary signal estimated from the revised disturbance estimator. Though we learn only the steady-state so that the learned model-plant mismatch map is not perfect, we can achieve the offset-free tracking property by utilizing the stabilizing property of the supplementary signal estimator. Naturally, we could further improve the closed-loop performance of the controller with the entirely learned

model-plant mismatch map, but it would be required as enormous data and computation as for learning the entire dynamics of the system. Therefore, only learning and utilizing a tiny manifold on the entire model-plant mismatch map, steady-state mismatch, we can efficiently improve the closed-loop performance of offset-free MPC.

In addition, we could mathematically examine the robust asymptotic stability of the combined system in offset-free MPC consisting of disturbance estimator/target calculator/optimizer. Though the stability analysis of offset-free MPC have not been done yet due to the difficulty in specifying the equilibrium point and Lyapunov function candidate with nominal scheme, the proposed scheme makes it possible by providing the equilibrium point and Lyapunov function candidate based on the learned model-plant mismatch.

Moreover, since we combine machine learning and model based control based on standard offset-free MPC manner, we can exploit its own model-plant mismatch compensating property in estimator design. Therefore, proposed method can effectively improve the MPC performance without enormous data, unlike existing schemes improving model-based control performance by updating entire dynamics or learning entire model-plant mismatch compensating signal directly.

In this study, we learn and utilize the model-plant mismatch at steady-state for each set-point in tracking problem, but there are also many regulation problems in chemical processes such as regulating the concentration or temperature of continuous stirred-tank reactor. In case of regulation problem, set-point change does not exist. However, if the change of measured disturbance exists, we can apply the proposed model-plant mismatch learning offset-free MPC scheme, since the influence of measured disturbance change is basically identical as

the influence of the set-point change to the system.

5.4 Conclusions

In this thesis, we improve the optimality performance of move blocked MPC, which fixes the decision variables over arbitrary time intervals to reduce computational load for on-line optimization in MPC, in two ways. The first scheme provides a superior base sequence by linearly interpolating complementary base sequences, and the second scheme provides a proper time-varying blocking structure with semi-explicit approach. In these days, though the computation power of computers is getting increasing, there always exist cases where the computation capability is limited, for instance, MPC is directly implemented in the on-board controller such as engine control units of cars or planes and embedded controller in chemical plants. Therefore, needs for computationally efficient control schemes always exists, and proposed move blocking MPC schemes can be valuable alternatives in these situations.

We also improve the optimality performance of disturbance estimator based offset-free MPC, which accomplish offset-free tracking in MPC with additional disturbance signal, by learning and utilizing the learned model-plant mismatch signal from steady-state disturbance data with the supplementary signal from disturbance estimator. This scheme also valuable in viewpoint of providing a successful and efficient combining method for the model and data based method. This scheme would be one of useful starting points of machine learning based MPC study.

5.5 Future work

In *Chapter 2*, we parameterized the input sequence in terms of deviations from the linear interpolation of the infinite-horizon LQR solution and the shifted version of previous optimal solution. We formulated the base sequence with linear interpolation because we can reach with both base sequence with only one additional variable, interpolation parameter. Additionally, if we have sufficient computation capability, we can assign weights for each shifted previous solution and LQR solution. By this, we can expand the space of constructible base sequence as the range space of two basis vector, shifted previous solution and LQR solution, and formulate the base sequence as the result from a linear operator with the assigned weights. Therefore, studying and developing the extended version of offset-blocked MPC based on this linear operator would be a useful research.

In *Chapter 4*, we learned and utilized the intrinsic model-plant mismatch based on only one model. However, when the operating range of the plant becomes extensive, since control performance can considerably degraded only with one model, we usually apply model-bank approach which builds multiple model and utilizes each model according to current state. In this case, we also have to learn multiple model-plant mismatch map for each model, since the intrinsic model-plant mismatch are different for each model. Therefore, when we change the model during operation, we also have to change the model-plant mismatch map and apply the different learned disturbance signal according to the current state. However, this procedure is not that simple. Since the learned signal is derived from the set-point, in case where the current point and set-point are in different regions,

we have to transform the learned disturbance signal from the region including the set-point into the suitable signal for the region including current point. Therefore, studying and developing the model-plant mismatch learning offset-free MPC for model-bank approach would be a valuable research.

Nomenclature

d: Disturbance

$e_{\bar{x},\bar{u}}$: Target error

$e_{\hat{x}_d}$: Estimate error of disturbance-augmented state

e_{κ_p}: Error in control law

$e_{y,\infty}$: Steady-state output prediction error

$e_{z,\infty}$: Steady-state offset of controlled variable

e_c: Combined error

e_y: Measurement error

f_d: Steady-state disturbance map from each set-point

g: Parameter vector

i^*: Corresponding index of $\widetilde{s}^*$

J: Objective function

J^*: Optimal objective value

k: Discrete-time index

L_d: Estimator gain for disturbance vector

L_x: Estimator gain for state vector

N: Prediction horizon

n_c: Number of inequalities

n_d: Dimension of disturbance vector

n_s: Number of admissible blocking position sets

n_u: Dimension of input vector

n_x: Dimension of state vector

n_y: Dimension of output vector

O_∞: Maximal positive invariant

P_s: Blocking matrix from blocking position set s

Q_N: Weighting matrix for the terminal state vector

Q_u: Weighting matrix for the input vector

Q_x: Weighting matrix for the state vector

r: Reference signal

s: Blocking position set

s_i: Blocking position

U: Future input sequence

u: Input

U^*: Optimal solution sequence

U_B: Base sequence

U^*_{k-1}: Optimal solution sequence of the previous sampling instant

U_{LQR}: LQR solution sequence

u_∞: Steady-state input

x: State

x_d: Disturbance-augmented state

x_p: Plant state

y: Output

$y_{p,\infty}$: steady-state plant output

y_p: Plant output

z_p: Plant controlled variable

$\bar{r}$: Set-point

$\bar{u}$: Target input

$\bar{x}$: Target state

$\bar{x}^{\ell^*}$: Target input derived from $\hat{d}^{\ell^*}$

$\bar{x}^{\ell^*}$: Target state derived from $\hat{d}^{\ell^*}$

$\Delta\Theta$: Offset variation sequence

Δu: Input variation

ΔU: Input variation sequence

$\Delta\overline{\Theta}$: Parameterized offset variation sequence

$\Delta\widetilde{\Theta}_s^*$: Unconstrained optimal offset variation sequence with blocking position s

$\hat{d}$: Disturbance estimate

$\hat{d}^{\ell^*}$: Learned disturbance signal from completely learned $\hat{f}_d$

$\hat{d}^{\ell}$: Learned disturbance signal

$\hat{d}^{s}$: Estimated supplementary disturbance signal

$\hat{d}_\infty$: Disturbance estimate at steady-state

$\hat{f}_d$: Approximated function of f_d

$\hat{x}$: State estimate

$\hat{x}^{\ell,s}$: Combined disturbance signal

$\hat{x}_\infty$: State estimate at steady-state

$\hat{x}_d$: Disturbance-augmented state estimate

$\hat{x}_d^+$: Expected evolved state with perturbation

κ_{un}: Unconstrained MPC controller gain

λ: Interpolation parameter

$\mathbb{Z}_{\geq 0}$: Set of non-negative integers

$\mathcal{C}$: Control invariant set

$\mathcal{C}_\infty$: Maximal control invariant set

$\mathcal{I}_{s(i)}^{dom}$: Set of indices j that form the dominant boundaries for $CR_{s(i)}$

$\mathcal{I}_s$: Set of indices for each admissible blocking position set

$\mathcal{N}$: Null space

$\mathcal{S}$: Set of admissible blocking position sets

$\mathcal{S}_N$: Collection of admissible s with $s_{\overline{N}} = N$

$\mathcal{X}_T$: Terminal constraint set

$\otimes$: Kronecker product

$\overline{\Theta}$: Parameterized offset sequence

$\overline{N}$: Number of blocks

$\overline{P}$: Blocking matrix

ϕ : Single-stage cost

ϕ_t: Terminal cost

Θ: Offset sequence

$\tilde{x}_d^+$: Expected ideally evolved state

$\widehat{U}$: Shifted version of previous solution

$\widetilde{J}_s^*$: Unconstrained value function with blocking position s

$\widetilde{s}^*$: Blocking position set minimizing $\widetilde{J}_s^*$

$\widetilde{s}^*$: Proper blocking position set

$B_{s(i,j)}$: Boundary between the critical regions $CR_{s(i)}$ and $CR_{s(j)}$

$CR_{s(i)}$: Critical region associated with $s(i)$

Bibliography

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[2] Y. Kim, T. Park, C. Jung, C. H. Kim, Y. W. Kim, and J. M. Lee, "Hybrid nonlinear model predictive control of lnt and urealess scr aftertreatment system," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2305–2313, 2019.

[3] B. S. Kim, T. Y. Kim, T. C. Park, and Y. K. Yeo, "A model predictive functional control based on proportional-integral-derivative (pid) and proportional-integral-proportional-derivative (pipd) using extended non-minimal state space: Application to a molten carbonate fuel cell process," *Korean Journal of Chemical Engineering*, vol. 35, no. 8, pp. 1601–1610, 2018.

[4] H. Shin, Y. K. Lim, S.-K. Oh, S. G. Lee, and J. M. Lee, "Dynamic matrix control applied on propane-mixed refrigerant liquefaction process," *Korean Journal of Chemical Engineering*, vol. 34, no. 2, pp. 287–297, 2017.

[5] M. Ataei, A. Khajepour, and S. Jeon, "Model predictive rollover prevention for steer-by-wire vehicles with a new rollover index," *International Journal of Control*, vol. 0, no. 0, pp. 1–16, 2018.

[6] Z. Li, W. Zhong, Y. Liu, N. Luo, and F. Qian, "Dynamic modeling and control of industrial crude terephthalic acid hydropurification process," *Korean Journal of Chemical Engineering*, vol. 32, no. 4, pp. 597–608, 2015.

[7] M. Robillart, P. Schalbart, F. Chaplais, and B. Peuportier, "Model reduction and model predictive control of energy-efficient buildings for electrical heating load shifting," *Journal of Process Control*, vol. 74, pp. 23–34, 2019.

[8] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, p. 415, 2011.

[9] S. Engell, "Feedback control for optimal process operation," *Journal of process control*, vol. 17, no. 3, pp. 203–219, 2007.

[10] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, 2002.

[11] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, p. 267, 2010.

[12] J. A. Rossiter and L. Wang, "Exploiting laguerre functions to improve the feasibility/performance compromise in mpc," in *2008 47th IEEE Conference on Decision and Control*, pp. 4737–4742, IEEE, 2008.

[13] B. Khan and J. A. Rossiter, "Alternative parameterisation within predictive control: a systematic selection," *International Journal of Control*, vol. 86, no. 8, pp. 1397–1409, 2013.

[14] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[15] J. A. Rossiter and Y. Ding, "Interpolation methods in model predictive control: an overview," *International Journal of Control*, vol. 83, no. 2, pp. 297–312, 2010.

[16] R. Gondhalekar, J.-i. Imura, and K. Kashima, "Controlled invariant feasibility—a general approach to enforcing strong feasibility in mpc applied to move-blocking," *Automatica*, vol. 45, no. 12, pp. 2869–2875, 2009.

[17] B. Kouvaritakis, J. A. Rossiter, and M. Cannon, "Linear quadratic feasible predictive control," *Automatica*, vol. 34, no. 12, pp. 1583–1592, 1998.

[18] J. A. Mendez, B. Kouvaritakis, and J. A. Rossiter, "State-space approach to interpolation in mpc," *International Journal of Robust and Nonlinear Control*, vol. 10, no. 1, pp. 27–38, 2000.

[19] J. A. Rossiter and P. Grieder, "Using interpolation to improve efficiency of multiparametric predictive control," *Automatica*, vol. 41, no. 4, pp. 637–643, 2005.

[20] J. A. Rossiter, M. J. Rice, J. Schuurmans, and B. Kouvaritakis, "A computationally efficient constrained predictive control law," in *1998 American Control Conference*, vol. 6, pp. 3281–3285, IEEE, 1998.

[21] J. A. Rossiter, B. Kouvaritakis, and M. Cannon, "Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality," *International Journal of Control*, vol. 74, no. 17, pp. 1678–1689, 2001.

[22] J. A. Rossiter, B. Kouvaritakis, and M. Bacic, "Interpolation based computationally efficient predictive control," *International Journal of Control*, vol. 77, no. 3, pp. 290–301, 2004.

[23] M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis, "General interpolation in mpc and its advantages," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1092–1096, 2003.

[24] C.-J. Ong and Z. Wang, "Reducing variables in model predictive control of linear system with disturbances using singular value decomposition," *Systems & Control Letters*, vol. 71, pp. 62–68, 2014.

[25] R. C. Shekhar and C. Manzie, "Optimal move blocking strategies for model predictive control," *Automatica*, vol. 61, pp. 27–34, 2015.

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

[27] R. Gondhalekar and J.-i. Imura, "Least-restrictive move-blocking model predictive control," *Automatica*, vol. 46, no. 7, pp. 1234–1240, 2010.

[28] J. A. Rossiter, B. Kouvaritakis, and M. J. Rice, "A numerically robust state-space approach to stable-predictive control strategies," *Automatica*, vol. 34, no. 1, pp. 65–73, 1998.

[29] B. Kouvaritakis and M. Cannon, *Model Predictive Control*. Springer, 2016.

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

[31] R. C. Shekhar and J. M. Maciejowski, "Robust variable horizon mpc with move blocking," *Systems & Control Letters*, vol. 61, no. 4, pp. 587–594, 2012.

[32] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European Control Conference*, pp. 502–510, IEEE, 2013.

[33] S.-K. Oh, B. J. Park, and J. M. Lee, "Point-to-point iterative learning model predictive control," *Automatica*, vol. 89, no. 6, pp. 135–143, 2018.

[34] J.-P. Corriou, *Process Control*. Springer, 2004.

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

[36] P. TøNdel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[37] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 2013.

[38] M. N. Zeilinger, C. N. Jones, and M. Morari, "Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[39] M. Jost and M. Mönnigmann, "Accelerating model predictive control by online constraint removal," in *52nd IEEE Conference on Decision and Control*, pp. 5764–5769, IEEE, 2013.

[40] G. Goebel and F. Allgöwer, "Semi-explicit mpc based on subspace clustering," *Automatica*, vol. 83, pp. 309–316, 2017.

[41] U. Jassmann, S. Dickler, J. Zierath, M. Hakenberg, and D. Abel, "Model predictive wind turbine control with move-blocking strategy for load alleviation and power leveling," in *Journal of Physics: Conference Series*, vol. 753, p. 052021, IOP Publishing, 2016.

[42] U. Z. Abdul Hamid, H. Zamzuri, T. Yamada, M. A. Abdul Rahman, Y. Saito, and P. Raksincharoensak, "Modular design of artificial potential field and nonlinear model predictive control for a vehicle collision avoidance system with move blocking strategy," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 10, pp. 1353–1373, 2018.

[43] S. Longo, E. C. Kerrigan, K. V. Ling, and G. A. Constantinides, "Parallel move blocking model predictive control," in *50th IEEE Conference on Decision and Control and European Control Conference*, pp. 1239–1244, IEEE, 2011.

[44] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit mpc," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 18, no. 8, pp. 816–830, 2008.

[45] I. S. Chin, K. S. Lee, and J. H. Lee, "A technique for integrated quality control, profile control, and constraint handling for batch processes," *Industrial & Engineering Chemistry Research*, vol. 39, no. 3, pp. 693–705, 2000.

[46] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 528–535, IEEE, 2016.

[47] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American Control Conference (ACC)*, pp. 1520–1527, IEEE, 2018.

[48] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[49] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, IEEE, 2017.

[50] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 39–45, IEEE, 2018.

[51] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit,"

Proceedings of the Royal Society A, vol. 474, no. 2219, p. 20180335, 2018.

[52] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018.

[53] I. Koryakovskiy, M. Kudruss, H. Vallery, R. Babuška, and W. Caarls, "Model-plant mismatch compensation using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2471–2477, 2018.

[54] E. Davison, "The robust control of a servomechanism problem for linear time-invariant multivariable systems," *IEEE transactions on Automatic Control*, vol. 21, no. 1, pp. 25–34, 1976.

[55] E. Davison and H. Smith, "Pole assignment in linear time-invariant multivariable systems with constant disturbances," *Automatica*, vol. 7, no. 4, pp. 489–498, 1971.

[56] C. Johnson, "Further study of the linear regulator with disturbances–the case of vector disturbances satisfying a linear differential equation," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 222–228, 1970.

[57] J. Pearson, R. Shields, and P. Staats, "Robust solutions to linear multivariable control problems," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 508–517, 1974.

[58] U. Maeder and M. Morari, "Offset-free reference tracking with model predictive control," *Automatica*, vol. 46, no. 9, pp. 1469–1476, 2010.

[59] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.

[60] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 2012.

[61] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.

[62] G. Pannocchia and A. Bemporad, "Combined design of disturbance model and observer for offset-free model predictive control," *IEEE Transactions on Automatic Control*, vol. 52, no. 6, pp. 1048–1053, 2007.

[63] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[64] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.

[65] G. Pannocchia, M. Gabiccini, and A. Artoni, "Offset-free mpc explained: novelties, subtleties, and applications," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 342–351, 2015.

[66] X. Sun, J. Liu, K. Zhu, J. Hu, X. Jiang, and Y. Liu, "Generalized regression neural network association with terahertz spectroscopy for quantitative analysis of benzoic acid additive in wheat flour," *Royal Society Open Science*, vol. 6, no. 7, p. 190485, 2019.

[67] R. Rooki, "Application of general regression neural network (grnn) for indirect measuring pressure loss of herschel–bulkley drilling fluids in oil drilling," *Measurement*, vol. 85, pp. 184–191, 2016.

[68] D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings, "On the inherent robustness of optimal and suboptimal nonlinear mpc," *Systems & Control Letters*, vol. 106, pp. 68–78, 2017.

[69] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[70] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.

[71] M. Lazar, W. P. M. H. Heemels, and A. R. Teel, "Further input-to-state stability subtleties for discrete-time systems," in *UKACC International Conference on Control 2010*, pp. 613–618, IET, 2010.

[72] H. K. Khalil, *Nonlinear Systems*. 2002.

[73] W. Rudin, *Principles of Mathematical Analysis*, vol. 3. McGraw-hill New York, 1964.

[74] J. B. Rawlings and M. J. Risbeck, "On the equivalence between statements with epsilon-delta and k-functions," tech. rep., Technical Report 2015–01, TWCCC Technical Report, December 2015. URL http://jbrwww.che.wisc.edu/tech-reports/twccc-2015-01.pdf, 2015.

초 록

모델예측제어는 현재 시스템 상태에 대한 유한 구간 최적해를 도출하는 온라인 이동 구간 제어 방식이다. 모델예측제어는 피드백을 통한 공정 동특성과 제약 조건을 효과적으로 반영하는 장점으로 인해 산업 및 제어 연구 분야에 큰 영향을 미쳤다. 이러한 모델예측제어에는 몇 가지 해결되어야 할 문제가 있다. 모델예측제어에서는 샘플링 기간 내에 최적화 문제를 풀어내야 하기 때문에, 온라인 계산 복잡성의 감소가 주요 연구 주제 중 하나로 활발히 연구되고 있다. 또 다른 주요 문제는 모델에 기반한 예측을 이용하는 접근 방식으로 인해 모델-플랜트 불일치로 인한 오차를 해결해야 한다는 점이며, 모델 플랜트 불일치 또는 측정되지 않은 외란을 보상하여 잔류편차 없이 참조신호를 추적하는 연구가 활발히 이루어지고 있다. 이 논문에서는 모델예측제어에서의 온라인 최적화를 위한 계산 부하를 줄이기 위해 임의의 시간 간격에 걸쳐 결정 변수를 고정시키는 이동 블록 전략의 최적성 향상에 중점을 두었으며, 또한 잔류편차를 제거하기 위해 가장 표준적으로 사용되는 외란 추정기를 이용한 잔류편차-제거 모델예측제어 기법의 최적성 향상에 중점을 두었다. 이 논문에서는 이동 블록 모델예측제어의 최적 성능을 향상시키기 위한 두 가지 전략을 제시한다. 첫 번째 전략은 이동 블록 전략에서 일반적으로 고정된 채로 사용되는 기반 시퀀스를 상호 보완적인 두 기반 시퀀스의 선형 보간으로 대체함으로써 보다 우수한 기반 시퀀스를 제공하며, 두 번째 전략은 준-명시적 접근법을 활용하여 현재 시스템 상태에 적절한 시변 블록 구조를 온라인에서 제공한다. 또한, 잔류편차-제거 모델예측제어 기법의 최적 성능을 향상시키기 위해 추정 외란 데이터로부터 학습된 모델-플랜트 불일치 보상 신호를 온라인에서 이용하는 전략을 제안하였다. 제안된

세 가지 기법을 통해 모델예측제어의 반복적 실현가능성과 폐쇄-루프 안정성을 보장하면서 최적 성능을 효율적으로 개선 하였다.

주요어 : 모델예측제어, 입력 파라미터화, 이동블록 전략, 모델 플랜트 불일치, 잔류편차 제거
학번 : 2016-30232