# Accelerated Termination in Dual Decomposition Based Iterations for Model Predictive Control

Xiang Dai

DOCTORAT BRETAGNE LOIRE / MATHSTIC

CentraleSupélec

# THÈSE DE DOCTORAT DE

CENTRALESUPÉLEC

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Automatique, Productique, Robotique*

Par
**Xiang DAI**

# Accelerated Termination in Dual Decomposition Based Iterations for Model Predictive Control

**Thèse présentée et soutenue à Rennes, le mercredi 7 juillet 2021**
**Unité de recherche : Institut d'Electronique et des Technologies du numéRique (IETR)**
**Thèse N° : 2021CSUP0007**

## Rapporteurs avant soutenance :

| | |
|---|---|
| Mazen ALAMIR | Professeur, Université Grenoble Alpes |
| Nicolas LANGLOIS | Professeur, ESIGELEC, Université de Rouen Normandie |

## Composition du Jury :

| | | |
|---|---|---|
| Président : | Pierre RIEDINGER | Professeur, Université de Lorraine |
| Examinateurs : | Mazen ALAMIR | Professeur, Université Grenoble Alpes |
| | Nicolas LANGLOIS | Professeur, ESIGELEC, Université de Rouen Normandie |
| | Pierre RIEDINGER | Professeur, Université de Lorraine |
| | Cristina MANIU | Professeur, CentraleSupélec |
| | Hervé GUEGUEN | Professeur, CentraleSupélec |
| | Romain BOURDAIS | Maître de conférence, CentraleSupélec |
| Dir. de thèse : | Hervé GUEGUEN | Professeur, CentraleSupélec |

# Contents

# Acknowledgement

First, I would like to express my boundless gratitude to my parents for their constant caring and backing during my entire Ph.D. program, which dates back to when I started applying for a Ph.D. position. This whole journey of my Ph.D. program would not even begin if it was not for their unconditional support and love.

I am extremely grateful to my Ph.D. supervisors, Prof. Romain Bourdais and Prof. Hervé Guéguen, for their rigorous guidance and sustained encouragement to my Ph.D. study. The quarantine at Wuhan of 9 months in 2020, which is caused by Covid-19, is the most challenging period of my Ph.D. During that time, Romain and Hervé have given me enormous support by sending warm greetings to my family and me countless times, directing me in revising articles through frequent video and telephone meetings, and encouraging me to be patient and confident in facing the tough pandemic situation. They have also offered me precious freedom through all these years, which has enabled me to develop the Ph.D. subject completely by following my interest and curiosity. That has led to many new ideas from unprecedented angles, systematic and theoretical exploration of new fields, I appreciate greatly the freedom given by Romain and Hervé in the first place. Specifically, I am deeply impressed by Romain's prompt feedback in reviewing my work and quick thinking in discussions of technical details, which empowered me in a dynamic environment to do research. And for Hervé, his prudence in examining mathematical proofs and high standards of clarity and compactness for article writing has laid a long-lasting influence on my Ph.D. study.

In addition, for other professors in our group, I want to send my acknowledgment: thank Prof. Stanislav Aranovskiy for his enthusiastic encouragement and meaningful advice on research and career path, thank Prof. Pierre Haessig for informing me about various programming skills and novel productivity tools, thank Prof. Marie-Anne Lefebvre and Prof. Nabil Sadou for their kind support and encouragement, thank Prof. Alexandros Charalampidis for worthy discussions about postdoc plan and choice. In the sequel, I would like to thank Prof. Carlos Bader and Prof. Stanislav Aranovskiy for serving as the chair and member of my "Comité de Suivi Individual", their periodical exam and constructive advice help to keep my Ph.D. on the right track.

For our group's Ph.D. and postdoc colleagues, Amanda Abreu, Jesse-James A. Prince Agbodjan, Ioannis Kordonis, Rafael Accacio Nogueira and Zhigang Zhang, I would like to express my gratitude for spending the memorable time together and the numerous practical tips they have shared.

I also gratefully acknowledge the administrative staff of CentraleSupélec, including Mme. Karine Bernard, Mme. Jeannine Hardy, Mme. Catherine Piednor, Mme. Cecile Dubois and M. Gregory Cadeau, for their generous help and assistance; the whole technical support department 5050 for their always quick response and professionalism in fixing hardware and software malfunctions.

Further, I am grateful to Mme. Sibel Kus, Mme. Tronel Jeuland, M. Chunqiao Wang, Dr. Baptiste Leroy, Dr. Gaosong Wu, M. Han Qin, Prof. Didier Mayer, Prof. Michel Aublant, Mme. Yang Liu, my master program supervisor Prof. Yankun Jiang, my family members and my friends in China and France, space limitation prohibits me from naming every one of them, yet their caring and support matter a great deal for my Ph.D. Particularly, I want to thank Lingxia Li for her selfless and immeasurable support over the last half-year, the most stressful period of my Ph.D. The completion of my Ph.D. program and this dissertation would not be possible without her backing.

At last, I would like to express my gratitude to l'Ambassade de Chine en France for the aid of life and study in France, and to my country for the support in every visible and invisible aspect.

# NOMENCLATURE

### Variables

| | |
|---|---|
| $x$ | State variables |
| $u$ | Input variables |
| $m$ | Number of subsystems |
| $N$ | Prediction horizon length |
| $\boldsymbol{x}$ | Vector aggregates state variables of all subsystems from step 1 to $N$ |
| $\boldsymbol{u}$ | Vector aggregates input variables of all subsystems from step 0 to $N-1$ |
| $\boldsymbol{y}$ | Vector aggregates $\boldsymbol{x}$ and $\boldsymbol{u}$ |
| $\epsilon$ | Suboptimality |
| $\boldsymbol{\lambda}$ | Dual variables associated with inequality constraints |
| $\boldsymbol{\theta}$ | Dual variable associated with equality constraints |
| $\alpha_{\boldsymbol{\theta}}$ | Step size in iterative process of $\boldsymbol{\theta}$ |
| $\alpha_{\boldsymbol{\lambda}}$ | Step size in iterative process of $\boldsymbol{\lambda}$ |
| $\Delta\boldsymbol{\theta}$ | Search direction in iterative process of $\boldsymbol{\theta}$ |
| $\Delta\boldsymbol{\lambda}$ | Search direction in iterative process of $\boldsymbol{\lambda}$ |

### Sets

| | |
|---|---|
| $\mathbb{R}$ | Real numbers |
| $\mathbb{S}_+, \mathbb{S}_{++}$ | Symmetric positive semidefinite, positive definite matrix |
| $\mathbb{N}$ | Natural numbers |

### Operators and functions

| | |
|---|---|
| $f : A \to B$ | $f$ is a function on its domain set, which is subset of $A$, into the set $B$ |

| | |
|---|---|
| $\nabla f$ | Gradient of function $f$ |
| $\nabla^2 f$ | Hessian of function $f$ |
| $Df$ | Derivative matrix of function $f$ |
| $\min \mathrm{eig}(A)$ | Minimal eigenvalue of matrix $A$ |
| $A \oplus B$ | $(A^T, B^T)^T$ , with matrix (or vectors) $A$ and $B$ possess the same column number |
| $\mathrm{rank}(A)$ | Rank of matrix $A$ |
| $\mathrm{blkdiag}(A, B)$ | Form a block matrix with the main-diagonal blocks as $A$ and $B$, and all off-diagonal blocks are zero matrices |
| $(\cdot)_a$ | affiliated matrix, vector, scalar, projection operator, set and variable at step $a$, $a \in \mathbb{N}$, $1 \leq a \leq N$ |
| $(\cdot)_{(a:b)}$ | affiliated matrix, vector, scalar, projection operator, set and variable from step $a$ to $b$, $a, b \in \mathbb{N}$ and $1 \leq a \leq b \leq N$ |

**Acronyms**

| | |
|---|---|
| MPC | Model Predictive Control |
| QP | Quadratic programming |
| mp-QP | multi-parametric Quadratic Programming |
| DLMFA | Dynamic Lagrange Multipliers Fixation Algorithm |
| LOPDSA | Local Optimization Problems Dynamic Sizing Algorithm |
| FPH-P | Full Prediction Horizon stopping condition with Projection |
| FS-P | First Step stopping condition with Projection |
| PDIPM | Primal Dual Interior Point Method |
| ATPDIPM | Accelerated Termination of Primal Dual Interior Point Method |
| LP | Linear Programming |
| CP | Cone Programming |
| LICQ | Linear Independence Constraint Qualification |

| | |
|---|---|
| ASBSA | Active Set Based Suboptimal Algorithm |
| POASIM | Proactive Optimal Active Set Identification Method |

# List of Figures

# List of Tables

# RÉSUMÉ ÉTENDU EN FRANÇAIS

## Contexte

La commande predictive (MPC) est une stratégie de contrôle optimal apparue à la fin des années 1970, issue de la commande algorithmique par modèle (MAC) [52] [53] et de la commande matricielle dynamique (DMC) [16]. Ses premières applications, qui ont vu le jour dans l'industrie des procédés chimiques dans les années 1980, sont dues à la simplicité de l'algorithme et à sa capacité à gérer des systèmes multivariables [22] [11]. Depuis lors, plusieurs étapes importantes ont été franchies dans le développement du MPC : contrôle adaptatif initié par la commande prédictive généralisée (GPC) [15] pour un processus mono variable, formulation de l'espace d'état pour la mise en œuvre d'un horizon décroissant [22], prise en compte de la faisabilité par les variables accessoires [72] et vérification associée [55], preuve de stabilité des MPC contraints [51] [54] [70] [34].

Grâce à la possibilité de traiter des dynamiques et des contraintes complexes [33] [10] [30] en fournissant une stratégie de contrôle optimale, la MPC a gagné en popularité dans de nombreuses industries au cours des dernières décennies, par exemple des applications étendues dans les industries de processus et chimiques [49], une croissance significative dans les industries aérospatiales et automobiles [11], application émergente dans la gestion de la chaîne logistique [14] [47], l'économie [18] [27] et la finance [48] [59].

La MPC génère une action de contrôle optimale pour l'instant présent tout en considérant l'influence des comportements futurs du système. Ceci est réalisé en produisant la séquence de contrôle optimale basée sur un modèle dynamique, visant à minimiser une fonction objectif, généralement quadratique, sur les prochaines étapes de contrôle (appelées horizon de prédiction).

Typiquement, à chaque pas de temps, un problème d'optimisation quadratique est formulé sur la base de l'état du pas courant, de la fonction objectif, du modèle dynamique, et des contraintes d'entrées et d'états (et de sorties). Une séquence de commande de la longueur de l'horizon de prédiction est obtenue en résolvant ce problème d'optimisation, et seules les entrées de la première étape sont appliquées au système. La formulation et le processus de résolution ci-dessus sont répétés après chaque intervalle d'échantillonnage, et

le nom de commande predictive provient donc de la caractéristique de prendre en compte les étapes temporelles futures du système contrôlé tout en décidant de l'action de contrôle de l'étape actuelle.

Aujourd'hui, l'enthousiasme de la recherche, qui a été lié à l'adoption industrielle des technologies dans les années 1980 et aux aspects théoriques, y compris l'interprétation de l'espace d'état et les preuves de stabilité dans les années 1990, s'oriente vers une mise en œuvre efficace et un calcul en ligne, c'est-à-dire comment résoudre efficacement l'optimisation quadratique sous contrainte.

Dans la pratique, cependant, la solution optimale du problème d'optimisation qui en résulte, avec des exigences de respect des contraintes tout en minimisant la fonction objectif, est parfois difficile à obtenir directement. Divers facteurs peuvent expliquer cela, par exemple, contraintes sur l'optimisation centralisée par une structure distribuée ou un souci de confidentialité, l'exigence d'un taux d'échantillonnage rapide par certaines applications, la limite de la mémoire ou de la puissance de calcul des unités de contrôle, pour n'en citer que quelques-uns. Parmi les méthodes efficaces de résolution du problème, la décomposition duale basée sur les multiplicateurs de Lagrange est depuis longtemps appréciée pour sa capacité à relaxer les contraintes et à intégrer le processus de résolution itératif qui en découle. Les facteurs critiques dans le processus itératif sont: la taille du pas, la direction de recherche et la condition d'arrêt.

La motivation principale de cette thèse est de générer une solution "suffisamment bonne" (répondant aux conditions d'arrêt prédéfinies ou aux exigences de sous-optimalité) de l'optimisation résultant de MPC plus rapide (terminaison accélérée du processus itératif basé sur la décomposition double), qui sera réalisée en exploitant les caractéristiques de la stratégie MPC et de la structure d'optimisation résultante MPC.

Dans la section suivante, le problème général d'optimisation résultant de MPC, son problème dual correspondant et le processus itératif seront formulés pour établir la base mathématique de cette thèse.

# Préliminaires mathématiques

Dans cette section, les préliminaires mathématiques fondamentaux du problème d'optimisation étudié dans cette thèse seront présentés, ce qui aidera le lecteur à former la base nécessaire pour les éléments présentés plus tard. Veuillez vous reporter à la

Nomenclature [1] pour les notations qui apparaissent dans cette section.

## La formulation générale du problème

Pour aborder la formulation du problème, le système à contrôler est d'abord présenté.

Dans cette thèse, le système linéaire invariant dans le temps considéré est composé de $m$ sous-systèmes avec un horizon de prédiction $N \in \mathbb{N}_{>0}$, et caractérisé par une dynamique de couplage comme suit :

$$x_l(j) = \sum_{i=1}^{m}(A_{li}x_i(j-1) + B_{li}u_i(j-1)),\ l = 1, ..., m,\ j = 1, ..., N, \tag{1}$$

où $x_l(j) \in \mathbb{R}^{nx_l}$ et $u_l(j-1) \in \mathbb{R}^{nu_l}$ sont les états à l'étape $j$ et les entrées à l'étape $j-1$ du $l$-ème sous-système, $A_{li} \in \mathbb{R}^{nx_l \times nx_i}$, $B_{li} \in \mathbb{R}^{nx_l \times nu_i}$ sont des matrices de système, et $x_l(0) = \bar{x}_l$, $\bar{x}_l \in \mathbb{R}^{nx_l}$ est l'état initial du $l$-ième sous-système.

Outre la dynamique du système, les contraintes d'égalité linéaires généralisées des variables d'entrée et d'état sont également considérées ci-dessous:

$$\sum_{l=1}^{m} A_j^l x_l(j) + \sum_{l=1}^{m} B_j^l u_l(j-1) = a_j, \tag{2}$$

où $A_j^l \in \mathbb{R}^{na_j \times nx_l}$, $B_j^l \in \mathbb{R}^{na_j \times nu_l}$ et $a_j \in \mathbb{R}^{na_j}$.

Dans la suite, des contraintes d'inégalité convexes sont imposées aux entrées et aux états du système comme:

$$f_i(\boldsymbol{x}, \boldsymbol{u}) \leq 0,\ i = 1, ..., n, \tag{3}$$

où $\boldsymbol{u} = (\boldsymbol{u}_0^T, ..., \boldsymbol{u}_{N-1}^T)^T$, $\boldsymbol{u} \in \mathbb{R}^{Nnu}$, $nu = nu_1 + ... + nu_m$, $\boldsymbol{x} = (\boldsymbol{x}_1^T, ..., \boldsymbol{x}_N^T)^T$, $\boldsymbol{x} \in \mathbb{R}^{Nnx}$, $nx = nx_1 + ... + nx_m$, pour $\forall\ j = 1, ..., N$, $\boldsymbol{u}_j = (u_1(j)^T, ..., u_m(j)^T)^T$, $\boldsymbol{u}_j \in \mathbb{R}^{nu}$, $\boldsymbol{x}_j = (x_1(j)^T, ..., x_m(j)^T)^T$, $\boldsymbol{x}_j \in \mathbb{R}^{nx}$, $f_i : \mathbb{R}^{Nnx} \times \mathbb{R}^{Nnu} \to \mathbb{R}$ est convexe, et $i$ désigne la $i$-ième contrainte d'inégalité, ce qui fait au total $n$ contraintes d'inégalité.

Le système mentionné ci-dessus est régi par un critère de MPC, ce qui conduit au

1. Les variables présentées dans la Nomenclature sont celles qui apparaissent dans le §1, dont les variations avec les accents sont utilisées dans les chapitres suivants pour différencier les significations dans des contextes de problèmes distincts, par exemple $\bar{\boldsymbol{\lambda}}$ et $\tilde{\boldsymbol{\lambda}}$ désignent les multiplicateurs de Lagrange associés aux contraintes d'inégalité dans le §2 et le §5 respectivement.

problème d'optimisation suivant:

$$J^* = \min_{\boldsymbol{u},\boldsymbol{x}} \; J(\boldsymbol{u},\boldsymbol{x}), \tag{4}$$
$$s.t.\ (1),(2),(3).$$

La fonction objectif quadratique basée sur la MPC du problème (4) est définie comme suit:

$$J(\boldsymbol{u},\boldsymbol{x}) = \sum_{j=1}^{N} J_j(\boldsymbol{u}_{j-1},\boldsymbol{x}_j),$$
$$J_j(\boldsymbol{u}_{j-1},\boldsymbol{x}_j) = \sum_{l=1}^{m} \frac{1}{2}(||u_l(j-1)||^2_{R^u_{lj}} + ||x_l(j)||^2_{R^x_{lj}}),$$

où $R^u_{lj} \in \mathbb{S}^{nu_l}_{++}$ et $R^x_{lj} \in \mathbb{S}^{nx_l}_{+}$ sont les matrices de pénalité des variables d'état et d'entrée respectivement.

Pour alléger la notation, le problème (4) est reformulé sous une forme compacte comme:

$$\mathcal{J}^* = \min_{\boldsymbol{y}} \; \mathcal{J}(\boldsymbol{y}), \tag{5a}$$
$$s.t.\ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \tag{5b}$$
$$\boldsymbol{f}(\boldsymbol{y}) \leq \boldsymbol{0}, \tag{5c}$$

où $\boldsymbol{y} \in \mathbb{R}^{ny}$, $ny = N(nx+nu)$, $\boldsymbol{y} = (\boldsymbol{y}_1^T,...,\boldsymbol{y}_N^T)^T$, $\boldsymbol{f} : \mathbb{R}^{ny} \to \mathbb{R}^n$ avec $\boldsymbol{f}(\boldsymbol{y}) = (\boldsymbol{f}_1(\boldsymbol{y})^T,...,\boldsymbol{f}_n(\boldsymbol{y})^T)^T$, et $\forall\ i = 1,...,n$, $\boldsymbol{f}_i(\boldsymbol{y}) = f_i(\boldsymbol{x},\boldsymbol{u})$, $\boldsymbol{y}_j = (\boldsymbol{x}_j^T,\boldsymbol{u}_{j-1}^T)^T$, $\mathcal{J}(\boldsymbol{y}) = \frac{1}{2}||\boldsymbol{y}||^2_{\boldsymbol{R}}$, $\boldsymbol{R} = \text{blkdiag}(\boldsymbol{R}_1,...,\boldsymbol{R}_N)$, $\boldsymbol{R} \in \mathbb{S}^{ny}_{+}$, $\boldsymbol{R}_j = \text{blkdiag}(\boldsymbol{R}^x_j,\boldsymbol{R}^u_j)$, $\boldsymbol{R}_j \in \mathbb{S}^{nx+nu}_{+}$, $\boldsymbol{R}^x_j = \text{blkdiag}(R^x_{1j},...,R^x_{mj})$, $\boldsymbol{R}^x_j \in \mathbb{S}^{nx}_{+}$, $\boldsymbol{R}^u_j = \text{blkdiag}(R^u_{1j},...,R^u_{mj})$, $\boldsymbol{R}^u_j \in \mathbb{S}^{nu}_{++}$.

L'expression de $\boldsymbol{A}$ et $\boldsymbol{b}$ peut être dérivée de (1) et (2) comme:

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_N \end{bmatrix},\ \boldsymbol{b}_1 = \begin{bmatrix} -\sum_{l=1}^{m} A_{1l}x_l(0) \\ \vdots \\ -\sum_{l=1}^{m} A_{ml}x_l(0) \\ a_1 \end{bmatrix}, \tag{6a}$$

$$\boldsymbol{b}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_j \end{bmatrix}, \text{ pour } j \neq 1, \tag{6b}$$

$$\boldsymbol{A} = \begin{bmatrix} B_1 & & & \\ A_2 & B_2 & & \\ & \ddots & \ddots & \\ & & A_N & B_N \end{bmatrix}, \tag{6c}$$

$$A_j = \begin{bmatrix} A_{11} & \dots & A_{1m} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \vdots & \vdots & & \vdots \\ A_{m1} & \dots & A_{mm} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}, \tag{6d}$$

$$B_j = \begin{bmatrix} -\boldsymbol{I} & & & B_{11} & \dots & B_{1m} \\ & \ddots & & \vdots & & \vdots \\ & & -\boldsymbol{I} & B_{m1} & \dots & B_{mm} \\ A_j^1 & \dots & A_j^m & B_j^1 & \dots & B_j^m \end{bmatrix}, \tag{6e}$$

où $\boldsymbol{b} \in \mathbb{R}^{n_e}$, $\boldsymbol{A} \in \mathbb{R}^{n_e \times ny}$, $n_e = Nnx + na_1 + ... + na_N$, $\forall\, j = 1, ..., N$, $A_j \in \mathbb{R}^{(nx+na_j)\times(nx+nu)}$, $B_j \in \mathbb{R}^{(nx+na_j)\times(nx+nu)}$, $\mathbf{0}$ et $\boldsymbol{I}$ sont respectivement une matrice d'éléments 0 et une matrice identité de taille propre.

## Problème dual et processus itératif général

Le cœur de cette thèse est de terminer plus rapidement le processus itératif basé sur la décomposition duale. Comme connaissances préalables, le problème dual de (5), et les hypothèses élémentaires sont présentés dans cette sous-section.

En utilisant la décomposition duale, le problème dual et le Lagrangien de (5) sont formulés comme:

$$g^* = \max_{\boldsymbol{\theta},\boldsymbol{\lambda}\geq 0} g(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\theta},\boldsymbol{\lambda}\geq 0} \min_{\boldsymbol{y}} \mathcal{L}(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}), \tag{7}$$

$$\mathcal{L}(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{2}||\boldsymbol{y}||^2_{\boldsymbol{R}} + \boldsymbol{\theta}^T(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{y}), \tag{8}$$

où $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$ et $\boldsymbol{\lambda} \in \mathbb{R}^n_+$ sont les variables duales associées à la contrainte (5b) et (5c)

respectivement.

**Hypothèse 1.** *La solution du problème* (5) *existe.*

Puisque $\mathcal{J}(\boldsymbol{y})$ est convexe, l'Hypothèse 1 est équivalente à la non vacuité de l'ensemble réalisable du problème (5).

**Hypothèse 2.** *L'intérieur de l'ensemble* $\{\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \boldsymbol{f}(\boldsymbol{y}) \leq \boldsymbol{0}\}$ *n'est pas vide, par exemple, il existe* $\boldsymbol{y} \in \mathbb{R}^{ny}$ *qui satisfait* $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$, *et* $\boldsymbol{f}(\boldsymbol{y}) < \boldsymbol{0}$.

**Remarque 1.** *Avec l'Hypothèse 2, la condition de Slater pour le problème* (5) *est satisfaite, de sorte que la dualité forte tient pour le problème* (4), *à savoir* $\mathcal{J}^* = g^*$.

En général, le problème (7) peut être résolu de manière itérative comme:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha_{\boldsymbol{\theta}}^k \Delta\boldsymbol{\theta}^k, \tag{9a}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha_{\boldsymbol{\lambda}}^k \Delta\boldsymbol{\lambda}^k, \tag{9b}$$

où l'exposant $k \in \mathbb{N}_+$ est le compteur d'itérations, $\alpha_{\boldsymbol{\theta}}^k, \alpha_{\boldsymbol{\lambda}}^k \in \mathbb{R}_{>0}$ sont les tailles de pas, $\Delta\boldsymbol{\theta}^k \in \mathbb{R}^{Nnx}$ et $\Delta\boldsymbol{\lambda}^k \in \mathbb{R}^{Nn}$ sont les directions de recherche associées à $\boldsymbol{\theta}^k$ et $\boldsymbol{\lambda}^k$ respectivement.

Notons que (7) formule simplement l'expression générale de $\boldsymbol{\lambda}$, dont la non négativité au cours de l'itération est assurée soit par $\max\{0, \boldsymbol{\lambda}^k + \alpha_{\boldsymbol{\lambda}}^k \Delta\boldsymbol{\lambda}^k\}$ ou un critère de sortie spécifique (sinon, $g(\boldsymbol{\theta}^k, \boldsymbol{\lambda}^k)$ est non borné ci-dessus), qui sera spécifié lorsque la méthode itérative déterminée est utilisée.

**Remarque 2.** *Soit* $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*)$ *la solution du problème* (7), *on sait [6] que sous certaines conditions de* $\alpha_{\boldsymbol{\theta}}^k$, $\alpha_{\boldsymbol{\lambda}}^k$ *(par exemple, règle de minimisation limitée, règle d'Armijo, etc.), et* $\Delta\boldsymbol{\theta}^k$, $\Delta\boldsymbol{\lambda}^k$ *(par exemple, direction réalisable, direction de descente, direction la plus raide, etc.), les séquences* $\{\boldsymbol{\theta}^k\}$ *et* $\{\boldsymbol{\lambda}^k\}$ *convergent vers* $\boldsymbol{\theta}^*$ *et* $\boldsymbol{\lambda}^*$ *respectivement.*

# Deux approches principales

En bref, les méthodes itératives et analytiques sont deux approches principales pour résoudre le problème d'optimisation convexe (5).

La méthode itérative a été étudiée pour la première fois en 1956 par Frank et Wolfe [20], et a progressé au cours des dernières décennies en combinaison avec la décomposition duale

et les méthodes d'optimisation non linéaire et convexe générale [41], qui se répartissent principalement en 2 classes: la méthode du premier ordre et la méthode du second ordre.

La méthode du premier ordre possède souvent un taux de convergence linéaire, dans laquelle le gradient ou le sous-gradient des fonctions objectif et contrainte est utilisé pour former la direction de recherche, notamment la méthode du gradient [2], la méthode du sous-gradient [6], la méthode du gradient conjugué [57], etc. En vertu de sa simplicité de mise en œuvre, de son insensibilité au changement de dimension et de la simplicité du calcul de la direction de recherche, la méthode du premier ordre est souhaitable pour les processus itératifs, en particulier pour les structures distribuées.

La méthode du second ordre, où le gradient du second ordre des fonctions objectif et contrainte est utilisé pour former la direction de recherche, a un taux de convergence beaucoup plus rapide (généralement quadratique) mais a une mise en œuvre plus compliquée, par exemple la méthode de Newton [29], la méthode quasi Newton [6], la méthode de Gauss Newton [64], etc. Sa convergence rapide, généralement de plusieurs dizaines d'itérations, peut atteindre une précision extrêmement élevée, ce qui lui confère une priorité élevée pour être employée à toutes sortes de problèmes d'optimisation convexe.

Les méthodes analytiques permettant de résoudre le problème (5) avec (5c) étant linéaires, dit optimisation quadratique linéaire sous contraintes, peuvent être classées en deux catégories : les méthodes numériques et les méthodes géométriques.

La méthode numérique, utilisant la programmation quadratique multiparamétrique (mp-QP) [1], a été initialement proposée dans [5], où l'état initial était considéré comme les multi-variables pour former une cartographie hors ligne en partitionnant son espace euclidien en régions critiques voisines. Cela s'obtient en deux étapes: premièrement, résoudre une programmation linéaire basée sur la condition de Karush-Kuhn-Tucker (KKT) pour un polyèdre avec un point de départ faisable donné; deuxièmement, visiter le côté opposé d'une frontière du polyèdre (hyperplan) une par une pour former d'autres régions critiques.

De nombreuses recherches, liées à la méthode numérique analytique, ont été développées soit pour étendre son champ d'application, soit pour améliorer son efficacité, notamment la réduction de la partition inutile de la région critique [61], les cas de qualification des contraintes d'indépendance non linéaires et les cas de hessian semi-défini [60], transfert optimal de l'ensemble actif dans l'horizon de prédiction $N-1$ à $N$ [37] [36], élagage de l'ensemble infaisable pendant la partition de la région critique [26], utilisation de la technique de traversée de graphe sans exigence de nondégénérescence [46]. Il convient de

noter qu'en pratique, visiter entièrement toutes les régions prend énormément de temps et épuise la mémoire, même pour un problème de taille moyenne [33], et la propriété "ergodique" de la visite de la région critique limite son applicabilité uniquement aux systèmes de petite taille [63].

La méthode géométrique, proposée dans [56], tire parti de la propriété géométrique des QP pour construire un ellipsoïde centré sur la solution optimale sans contrainte, grâce auquel la solution souhaitée est trouvée au point le plus proche dans chaque région partitionnée par l'hyperplan et ses normales. Cependant, cette méthode est limitée aux cas de contraintes en boîte.

Du §2 au §5, plusieurs méthodes différentes, dont la méthode du premier ordre, la méthode du second ordre et la méthode analytique numérique, seront appliquées pour résoudre le problème (5), en fonction des différentes configurations et hypothèses. Une revue de littérature plus détaillée sera donnée au début de chaque chapitre.

# La motivation et principales contributions

## La motivation

Un fait universel mais pessimiste concernant presque tous les types de méthodes itératives est que l'optimalité et la faisabilité, en termes de problème primaire (5), ne sont garanties que dans la limite des itérations lorsque (9) est appliqué [8]. Par conséquent, d'un point de vue arithmétique (le processus itératif doit se terminer dans un nombre fini d'itérations, sinon aucune solution n'est atteinte) et pratique (toutes les simulations et applications préfèrent que le temps de résolution soit le plus court possible), il est obligatoire d'appliquer une condition d'arrêt lors de la mise en œuvre de (9).

Puisque la solution obtenue par une telle condition n'est pas assurée d'être optimale, la motivation de cette thèse est d'accélérer la fin du processus itératif (9) et de fournir des solutions qui peuvent satisfaire les critères prédéterminés, ce qui est réalisé en exploitant la structure temporelle du problème (5) montré dans (6) et la caractéristique MPC que seule la première étape de la séquence de contrôle est appliquée au système.

## Les principales contributions

Les principales contributions de cette thèse sont énumérées ci-dessous:

1. des nouveaux traitements pour accélérer la terminaison du processus itératif, y compris 2 algorithmes proposés dans le §2 pour réduire la complexité et améliorer l'efficacité pendant les itérations ;
2. des nouvelles conditions d'arrêt pour garantir la sous-optimalité, notamment la condition d'arrêt basée sur le gradient proposée dans le §3, et la condition d'arrêt basée sur l'identification de l'ensemble actif proposée dans le §5, où la preuve mathématique de la borne inférieure de la sous-optimalité est donnée ;
3. des nouveaux traitements pour assurer la faisabilité, notamment le mécanisme de projection proposé dans le §3, l'approche basée sur la programmation conique proposée dans le §5 ;
4. des conditions d'arrêt améliorées avec garantie de sous-optimalité et de faisabilité pour MPC (la caractéristique MPC selon laquelle seuls les composants de la première étape de la séquence de contrôle sont appliqués au système est exploitée), y compris la condition d'arrêt basée sur le gradient proposée dans le §3, et le critère basé sur la condition KKT modifiée proposé dans le §4.

# Le contour

Après avoir présenté l'introduction générale de la thèse dans ce chapitre, les techniques de réduction des itérations requises dans une MPC distribuée sont illustrées dans le §2, où la limite de sous-optimalité est démontrée pour deux nouveaux algorithmes proposés. Dans le §3, les conditions d'arrêt de sous-optimalité basées sur le gradient sont construites pour l'horizon de prédiction complet, et une projection ciblée de première étape est proposée pour produire une solution faisable et garantie de sous-optimalité. Dans le §4, la condition d'arrêt de la sous-optimalité de la première étape basée sur la méthode de point intérieur dual primal est démontrée pour le problème MPC avec des contraintes d'inégalité convexes générales. Le §5, qui sort un peu du cadre de MPC, décrit la situation de l'optimisation quadratique générale avec contraintes linéaires. Dans le §5, une condition de sous-optimalité intégrant la programmation du cône est conçue pour générer des solutions réalisables avec une garantie de sous-optimalité, ce qui permet une terminaison plus rapide du processus itératif. Une feuille de route est représentée à la Fig. 1, où une image globale de l'itinéraire et des caractéristiques de l'avancement de chaque chapitre est donnée.

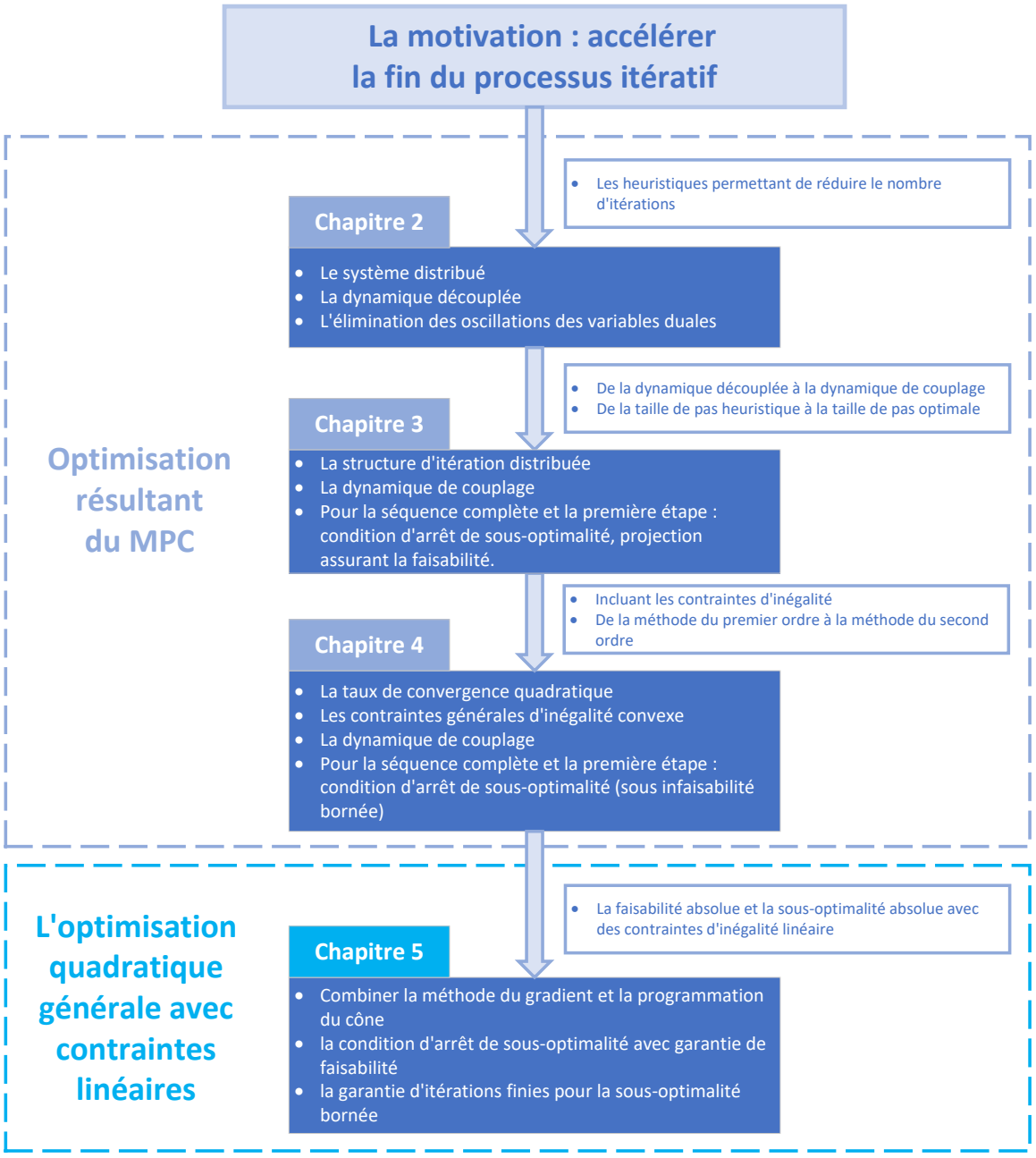


Figure 1 – La structure et la feuille de route technique de la thèse
Les flèches vers le bas indiquent le sens de l'évolution du chapitre précédent (ou heuristique générale) vers le chapitre suivant. La case à droite de chaque flèche donne la principale amélioration ou généralisation du chapitre ci-dessous par rapport au chapitre précédent. La case remplie en bleu sous chaque chapitre indique les principales caractéristiques du problème ou de la méthode qui y figurent.

Dans cette thèse, comme chaque chapitre s'attaque à une forme spécifique de problème (5) (caractérisé par une (5b) ou (5c) plus simplifiée ou plus généralisée) en employant différentes méthodes (méthode du premier ordre ou du second ordre, structure centralisée ou distribuée), la brève introduction de chaque chapitre est présentée ici, y compris la différenciation du problème, le savoir-faire de la technique proposée et les contributions exactes. Ainsi, le lecteur peut avoir un aperçu de l'essence de chaque chapitre sans avoir à creuser dans les détails.

## §2: La réduction dynamique de l'exigence d'itérations dans une MPC distribuée

Le §2 traite d'un cas particulier du problème (5), où aucune interaction de la dynamique entre les sous-systèmes n'est considérée, c'est-à-dire que la dynamique est découplée. En outre, les contraintes d'inégalité rencontrées dans le §2 sont une forme spécialisée de (5c), qui se compose de contraintes d'inégalité locales et globales. Ces dernières sont en outre considérées comme étant séparées en composantes de chaque sous-système.

Le §2 commence par une initiative visant à réduire le nombre d'itérations dans la résolution du problème résultant de MPC distribuée. À cette fin, l'algorithme de fixation dynamique des multiplicateurs de Lagrange (DLMFA) est proposé en fixant continuellement la valeur des multiplicateurs de Lagrange, et l'algorithme de dimensionnement dynamique des problèmes d'optimisation locale (LOPDSA) est proposé en réduisant continuellement la taille du problème d'optimisation locale pendant les itérations par une réduction de l'horizon de prédiction original. Les algorithmes proposés améliorent les performances de la méthode Uzawa en exploitant les contraintes séparables par étapes dans le contexte MPC. Ces améliorations découlent du comportement particulier des multiplicateurs de Lagrange et de leurs fluctuations sur l'horizon de prédiction. Des expériences numériques montrent que le nombre d'itérations et le temps de calcul de LOPDSA sont considérablement réduits par rapport à la méthode d'Uzawa.

## §3: La terminaison accélérée basée sur la $\epsilon$ sous-optimalité pour la MPC avec contraintes d'égalité

Le §3 est principalement basé sur la publication [17]. Dans le §3, le cas sans contrainte d'inégalité du problème (4) est considéré.

La décomposition duale est un outil efficace pour traiter les MPC problèmes, en particulier pour les MPC distribuées. Dans le §3, la limitation du nombre d'itérations est proposée en arrêtant le processus itératif une fois que la solution est suffisamment proche de la solution optimale. Les concepts de sous-optimalité primale et duale sont introduits, et les mécanismes de projection et les conditions d'arrêt sont dérivés, respectivement. En exploitant la structure particulière du problème MPC, où seules les entrées de la première étape sont appliquées au système, une condition d'arrêt $\epsilon$ sous-optimale plus rapide est conçue, se concentrant sur les composants uniquement à la première étape de l'horizon de prédiction, réduisant ainsi davantage le nombre d'itérations nécessaires. Au-delà des preuves théoriques développées, l'efficacité de la méthode, tant en temps de calcul qu'en nombre d'itérations, est illustrée par diverses simulations.

## §4: La terminaison accélérée basée sur la $\epsilon$ sous-optimalité pour la MPC en utilisant la méthode primale duale de points intérieurs

Dans le §4, on considère une structure séparable par étapes des contraintes d'inégalité (1.3).

L'utilisation de la méthode primale duale de points intérieurs pour le problème généré par la MPC afin de résoudre une solution sous-optimale est une approche mature avec des performances satisfaisantes. Dans le §4, la structure temporelle de la dynamique et des contraintes d'inégalité du système est exploitée. Un critère d'arrêt axé sur la première étape avec la garantie d'une sous-optimalité prédéfinie est proposé. La méthode qui intègre ce nouveau critère est supérieure en nombre d'itérations à la méthode de points intérieurs primale et duale existante. En plus des preuves mathématiques fournies, diverses simulations illustrent l'efficacité et l'efficience de la méthode.

## §5: La terminaison accélérée basée sur la $\epsilon$ sous-optimalité pour l'optimisation quadratique avec contraintes linéaires

Dans le §5, une forme plus généralisée de (5b) est considérée, où aucune dynamique d'états ou d'entrées n'est spécifiée, mais où les contraintes d'égalité linéaires générales sont imposées aux variables. En particulier, pour les contraintes d'inégalité (3), la forme linéaire générale non structurée est considérée.

De nombreuses méthodes utilisent la décomposition duale pour résoudre l'optimisation quadratique avec contraintes linéaire. Un mécanisme itératif assure la convergence vers

la solution optimale dans toutes ces méthodes. Cependant, la convergence est seulement garantie dans la limite des itérations. Dans le §5, un degré de sous-optimalité est introduit pour terminer le processus itératif plus rapidement tout en assurant le respect des contraintes. Une des clés principales de ce travail est l'identification des contraintes d'inégalités actives pendant le processus itératif. En plus des preuves mathématiques fournies, diverses simulations illustrent l'efficacité de la méthode proposée.

# Conclusion

Sous la condition d'arrêt par étapes dans une MPC distribuée, les deux algorithmes proposés dans le §2 peuvent réduire le nombre d'itérations requises en fixant la valeur des multiplicateurs de Lagrange et en abandonnant les étapes satisfaisantes dans le problème de la MPC, respectivement.

Dans le cadre d'une MPC distribuée avec contraintes d'égalité et dynamique de couplage, l'algorithme de projection à horizon de prédiction complet proposé dans le §3 possède un critère intégré basé sur le gradient et un mécanisme de projection pour garantir la sous-optimalité et la faisabilité. Un algorithme de projection focalisé sur la première étape avec garantie de sous-optimalité et de faisabilité a été proposé dans le §3, qui peut réduire considérablement le nombre d'itérations et le temps de calcul en répondant à la même exigence de sous-optimalité.

Dans le cadre d'une MPC avec une dynamique de couplage et des contraintes d'inégalité convexes séparables par étapes, le critère ciblé de première étape dans le respect de la sous-optimalité sous infaisabilité bornée pour la méthode de points intérieurs duale primale a été proposé dans le §4. L'algorithme qui en résulte s'est avéré supérieur en ce qui concerne le nombre d'itérations nécessaires, tant sur le plan théorique qu'expérimental.

Pour l'optimisation quadratique avec constraintes linéaires (pas nécessairement MPC), en combinant la technique d'identification de l'ensemble actif et la méthode du gradient général, une méthode proactive a été proposée pour fournir la solution optimale dans le §5. Un algorithme sous-optimal a été proposé dans le §5 basé sur la programmation du cône pour accélérer la fin du processus itératif en générant des solutions réalisables avec une sous-optimalité garantie. La borne inférieure de la sous-optimalité a également été démontrée.

Les travaux futurs concernant les algorithmes et les techniques proposés dans cette thèse peuvent être abordés dans les directions suivantes : étendre l'application des tech-

niques proposées, étudier en profondeur les propriétés des algorithmes proposés dans la théorie du contrôle et de l'optimisation, combiner les techniques proposées avec de nouvelles méthodes.

# Les publications

## Publié

— Xiang Dai, Romain Bourdais, and Hervé Guéguen, «Dynamic Reduction of the Iterations Requirement in a Distributed M odel Predictive Control», in: 2019IEEE 58th Conference on Decision and Control(CDC), IEEE, 2019, pp.6392–6397

## Soumis

— Xiang Dai, Romain Bourdais, and Hervé Guéguen, «$\epsilon$ Suboptimality Based Early Stop in Dual Decomposition for Model Predictive Control», Automatica, soumis en Octobre, 2020

## En préparation

— Xiang Dai, Romain Bourdais, and Hervé Guéguen, «$\epsilon$ suboptimality based accelerated termination for MPC using primal dual interior point method»

— Xiang Dai, Romain Bourdais, and Hervé Guéguen, «$\epsilon$ suboptimality based accelerated termination for linear constrained quadratic optimization»

Chapter 1

# Introduction

## 1.1 Background

Model predictive control (MPC) is an optimal control strategy arisen in late 1970s originated from Model Algorithmic Control (MAC) [52] [53] and Dynamic Matrix Control (DMC) [16]. Its early applications, emerged in chemical process industries in 1980s, are due to its simplicity of the algorithm and ability to handle multi variables [22] [11]. Several notable milestones in MPC developments since then are: adaptive control initiated Generalized Predictive Control (GPC) [15] for mono variable process, state space formulation for receding horizon implementation [22], feasibility circumvention through slack variables [72] and related verification [55], stability proof of constrained MPC controllers [51] [54] [70] [34].

Thanks to the capability of handling complex dynamics and constraints [33] [10] [30] in delivering optimal control strategy, MPC has gained increasing popularity in massive industries over the last few decades, e.g. expansive applications in process and chemical industries [49], significant growth in aerospace and automotive industries [11], emerging application in supply chain management [14] [47], economics [18] [27] and finance [48] [59].

MPC generates optimal control action for the current time instant while considering the influence of future behaviors of the system. This is achieved by producing the optimal control sequence based on a dynamic model, aiming at minimizing an objective function, usually quadratic, over the next several control steps (called prediction horizon).

At each time step, typically, a quadratic optimization problem is formulated based on the state of the current step, objective function, dynamic model, and constraints of inputs and states (and outputs). A control sequence of prediction horizon length is obtained by solving such an optimization problem, and only the first step inputs are applied to the system. The above formulation and solving process are repeated after each sampling interval, and the name of model predictive control is thus originated from the feature of taking into account the future time steps of the controlled system while deciding the

control action of the current step.

Today, the research enthusiasm, which has been granted to industrial adoption of the technologies in the 1980s and theoretical aspects including state space interpretation and stability proofs in the 1990s, is shifted towards efficient implementation and online computation [30], which is how to solve the constrained quadratic optimization efficiently.

In practice, however, the optimal solution for the resulting optimization problem, with requirements to fulfill constraints while minimizing the objective function, is sometimes intractable to obtain smoothly. Various factors may account for that, e.g., prohibition of centralized optimization manner by distributed structure or confidentiality concern, requirement of rapid sampling rate by certain applications, the limit of memory or computing power of control units, to name a few. Among the efficient methods to solve the problem, Lagrange multipliers based dual decomposition has long been appealing for its capability in relaxing constraints and integrating the subsequent accessible iterative solving process. The critical factors in the iterative process are step size, search direction, and stopping condition.

The primary motivation of this dissertation is to generate "good enough" solution (meeting the predefined stopping conditions or suboptimality requirements) of optimization resulted from MPC faster (accelerated termination of dual decomposition based iterative process), which will be realized by exploiting the features of MPC strategy and MPC optimization structure.

In the following section, the general MPC resulting optimization problem, its corresponding dual problem, and the iterative process will be formulated to establish the mathematical foundation of this dissertation.

## 1.2 Mathematical preliminaries

In this section, the mathematical preliminaries of the MPC optimization will be presented, which will help the reader to form the necessary basis for the materials presented later. Please refer to Nomenclature [1] for notations that appeared in this section.

1. The variables presented in Nomenclature are those appeared in §1, whose variations with accents are used in later chapters to differentiate the meanings under distinct problem settings, e.g. $\bar{\boldsymbol{\lambda}}$ and $\tilde{\boldsymbol{\lambda}}$ denote Lagrange multipliers associated with inequality constraints in §2 and §5 respectively.

### 1.2.1 General problem formulation

To bring up the problem formulation, the system to be controlled is first presented.

In this dissertation, the linear time invariant system considered is composed of $m$ subsystems with prediction horizon $N \in \mathbb{N}_{>0}$, and characterized by coupling dynamics as follow:

$$x_l(j) = \sum_{i=1}^{m}(A_{li}x_i(j-1) + B_{li}u_i(j-1)),\ l = 1, ..., m,\ j = 1, ..., N, \tag{1.1}$$

where $x_l(j) \in \mathbb{R}^{nx_l}$ and $u_l(j-1) \in \mathbb{R}^{nu_l}$ are states at step $j$ and inputs at step $j-1$ of $l$-th subsystem, $A_{li} \in \mathbb{R}^{nx_l \times nx_i}$ and $B_{li} \in \mathbb{R}^{nx_l \times nu_i}$ are system matrix, and $x_l(0) = \bar{x}_l$, $\bar{x}_l \in \mathbb{R}^{nx_l}$ is the initial state of $l$-th subsystem.

Aside from system dynamics, the generalized linear equality constraints of input and state variables are also considered below:

$$\sum_{l=1}^{m} A_j^l x_l(j) + \sum_{l=1}^{m} B_j^l u_l(j-1) = a_j, \tag{1.2}$$

where $A_j^l \in \mathbb{R}^{na_j \times nx_l}$, $B_j^l \in \mathbb{R}^{na_j \times nu_l}$ and $a_j \in \mathbb{R}^{na_j}$.

In the sequel, convex inequality constraints are imposed on system inputs and states as:

$$f_i(\boldsymbol{x}, \boldsymbol{u}) \leq 0,\ i = 1, ..., n, \tag{1.3}$$

where $\boldsymbol{u} = (\boldsymbol{u}_0^T, ..., \boldsymbol{u}_{N-1}^T)^T$, $\boldsymbol{u} \in \mathbb{R}^{Nnu}$, $nu = nu_1 + ... + nu_m$, $\boldsymbol{x} = (\boldsymbol{x}_1^T, ..., \boldsymbol{x}_N^T)^T$, $\boldsymbol{x} \in \mathbb{R}^{Nnx}$, $nx = nx_1 + ... + nx_m$, for $\forall\ j = 1, ..., N$, $\boldsymbol{u}_j = (u_1(j)^T, ..., u_m(j)^T)^T$, $\boldsymbol{u}_j \in \mathbb{R}^{nu}$, $\boldsymbol{x}_j = (x_1(j)^T, ..., x_m(j)^T)^T$, $\boldsymbol{x}_j \in \mathbb{R}^{nx}$, $f_i : \mathbb{R}^{Nnx} \times \mathbb{R}^{Nnu} \to \mathbb{R}$ is convex, and $i$ denotes the $i$-th inequality constraint, making in total $n$ inequality constraints.

The above mentioned system is governed by a MPC criterion, leading to the following optimization problem:

$$\begin{aligned} J^* &= \min_{\boldsymbol{u},\boldsymbol{x}}\ J(\boldsymbol{u}, \boldsymbol{x}), \\ s.t.\ &(1.1), (1.2), (1.3). \end{aligned} \tag{1.4}$$

The MPC based quadratic objective function of problem (1.4) is defined as:

$$J(\boldsymbol{u}, \boldsymbol{x}) = \sum_{j=1}^{N} J_j(\boldsymbol{u}_{j-1}, \boldsymbol{x}_j),$$
$$J_j(\boldsymbol{u}_{j-1}, \boldsymbol{x}_j) = \sum_{l=1}^{m} \frac{1}{2}(||u_l(j-1)||^2_{R^u_{lj}} + ||x_l(j)||^2_{R^x_{lj}}),$$

where $R^u_{lj} \in \mathbb{S}^{nu_l}_{++}$ and $R^x_{lj} \in \mathbb{S}^{nx_l}_{+}$ are penalty matrix of state and input variables respectively.

To lighten the notation, problem (1.4) is reformulated in a compact form as:

$$\mathcal{J}^* = \min_{\boldsymbol{y}} \ \mathcal{J}(\boldsymbol{y}), \tag{1.5a}$$
$$s.t. \ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \tag{1.5b}$$
$$\boldsymbol{f}(\boldsymbol{y}) \leq \boldsymbol{0}, \tag{1.5c}$$

where $\boldsymbol{y} \in \mathbb{R}^{ny}$, $ny = N(nx + nu)$, $\boldsymbol{y} = (\boldsymbol{y}_1^T, ..., \boldsymbol{y}_N^T)^T$, $\boldsymbol{f} : \mathbb{R}^{ny} \to \mathbb{R}^n$ with $\boldsymbol{f}(\boldsymbol{y}) = (\boldsymbol{f}_1(\boldsymbol{y})^T, ..., \boldsymbol{f}_n(\boldsymbol{y})^T)^T$, and $\forall\ i = 1, ..., n$, $\boldsymbol{f}_i(\boldsymbol{y}) = f_i(\boldsymbol{x}, \boldsymbol{u})$, $\boldsymbol{y}_j = (\boldsymbol{x}_j^T, \boldsymbol{u}_{j-1}^T)^T$, $\mathcal{J}(\boldsymbol{y}) = \frac{1}{2}||\boldsymbol{y}||^2_{\boldsymbol{R}}$, $\boldsymbol{R} = \text{blkdiag}(\boldsymbol{R}_1, ..., \boldsymbol{R}_N)$, $\boldsymbol{R} \in \mathbb{S}^{ny}_{+}$, $\boldsymbol{R}_j = \text{blkdiag}(\boldsymbol{R}^x_j, \boldsymbol{R}^u_j)$, $\boldsymbol{R}_j \in \mathbb{S}^{nx+nu}_{+}$, $\boldsymbol{R}^x_j = \text{blkdiag}(R^x_{1j}, ..., R^x_{mj})$, $\boldsymbol{R}^x_j \in \mathbb{S}^{nx}_{+}$, $\boldsymbol{R}^u_j = \text{blkdiag}(R^u_{1j}, ..., R^u_{mj})$, $\boldsymbol{R}^u_j \in \mathbb{S}^{nu}_{++}$.

The expression of $\boldsymbol{A}$ and $\boldsymbol{b}$ can be derived from (1.1) and (1.2) as:

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_N \end{bmatrix}, \ \boldsymbol{b}_1 = \begin{bmatrix} -\sum_{l=1}^{m} A_{1l} x_l(0) \\ \vdots \\ -\sum_{l=1}^{m} A_{ml} x_l(0) \\ a_1 \end{bmatrix}, \tag{1.6a}$$

$$\boldsymbol{b}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_j \end{bmatrix}, \ \text{for } j \neq 1, \tag{1.6b}$$

$$\boldsymbol{A} = \begin{bmatrix} B_1 & & & \\ A_2 & B_2 & & \\ & \ddots & \ddots & \\ & & A_N & B_N \end{bmatrix}, \tag{1.6c}$$

$$A_j = \begin{bmatrix} A_{11} & \dots & A_{1m} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \vdots & \vdots & & \vdots \\ A_{m1} & \dots & A_{mm} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}, \tag{1.6d}$$

$$B_j = \begin{bmatrix} -\boldsymbol{I} & & & B_{11} & \dots & B_{1m} \\ & \ddots & & \vdots & & \vdots \\ & & -\boldsymbol{I} & B_{m1} & \dots & B_{mm} \\ A_j^1 & \dots & A_j^m & B_j^1 & \dots & B_j^m \end{bmatrix}, \tag{1.6e}$$

where $\boldsymbol{b} \in \mathbb{R}^{n_e}$, $\boldsymbol{A} \in \mathbb{R}^{n_e \times ny}$, $n_e = Nnx + na_1 + ... + na_N$, $\forall\ j = 1, ..., N$, $A_j \in \mathbb{R}^{(nx+na_j)\times(nx+nu)}$, $B_j \in \mathbb{R}^{(nx+na_j)\times(nx+nu)}$, $\mathbf{0}$ and $\boldsymbol{I}$ are matrix of elements 0 and identity matrix of proper size respectively.

### 1.2.2 Dual problem and general iterative process

The core of this dissertation is to terminate faster the iterative process based on dual decomposition. As prerequisite knowledge, the dual problem of (1.5), and elementary assumptions are presented in this subsection.

Using dual decomposition, the dual problem and Lagrangian of (1.5) are formulated as:

$$g^* = \max_{\boldsymbol{\theta},\boldsymbol{\lambda}\geq 0} g(\boldsymbol{\theta},\boldsymbol{\lambda}) = \max_{\boldsymbol{\theta},\boldsymbol{\lambda}\geq 0} \min_{\boldsymbol{y}} \mathcal{L}(\boldsymbol{y},\boldsymbol{\theta},\boldsymbol{\lambda}), \tag{1.7}$$

$$\mathcal{L}(\boldsymbol{y},\boldsymbol{\theta},\boldsymbol{\lambda}) = \frac{1}{2}||\boldsymbol{y}||_{\boldsymbol{R}}^2 + \boldsymbol{\theta}^T(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{y}), \tag{1.8}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$ and $\boldsymbol{\lambda} \in \mathbb{R}_+^n$ are the dual variables associated with constraint (1.5b) and (1.5c) respectively.

**Assumption 1.** *It is assumed that the solution of problem* (1.5) *exists.*

Since $\mathcal{J}(\boldsymbol{y})$ is convex, Assumption 1 is equivalent to non emptiness of feasible set of problem (1.5).

**Assumption 2.** *It is assumed that the interior of set* $\{\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \boldsymbol{f}(\boldsymbol{y}) \leq \mathbf{0}\}$ *is not empty, e.g., there exists* $\boldsymbol{y} \in \mathbb{R}^{ny}$ *that satisfies* $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$ *and* $\boldsymbol{f}(\boldsymbol{y}) < \mathbf{0}$.

**Remark 1.** *With Assumption 2, the Slater's condition for problem* (1.5) *is satisfied, such that the strong duality holds for problem* (1.4)*, namely* $\mathcal{J}^* = g^*$.

In general, problem (1.7) can be typically solved in a iterative manner as:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha_{\boldsymbol{\theta}}^k \Delta\boldsymbol{\theta}^k, \tag{1.9a}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha_{\boldsymbol{\lambda}}^k \Delta\boldsymbol{\lambda}^k, \tag{1.9b}$$

where the superscript $k \in \mathbb{N}_+$ is iteration counter, $\alpha_{\boldsymbol{\theta}}^k, \alpha_{\boldsymbol{\lambda}}^k \in \mathbb{R}_{>0}$ are step size, $\Delta\boldsymbol{\theta}^k \in \mathbb{R}^{Nnx}$ and $\Delta\boldsymbol{\lambda}^k \in \mathbb{R}^{Nn}$ are search direction associated with $\boldsymbol{\theta}^k$ and $\boldsymbol{\lambda}^k$ respectively.

Note that (1.7) simply formulates the general expression of $\boldsymbol{\lambda}$, whose non negativeness during the iteration is ensured either by $\max\{0, \boldsymbol{\lambda}^k + \alpha_{\boldsymbol{\lambda}}^k \Delta\boldsymbol{\lambda}^k\}$ or specific quit criterion (otherwise, $g(\boldsymbol{\theta}^k, \boldsymbol{\lambda}^k)$ is unbounded above), which will be specified when determinate iterative method is used.

**Remark 2.** *Denote $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*)$ the solution of problem (1.7), it is known [6] that under certain conditions of $\alpha_{\boldsymbol{\theta}}^k, \alpha_{\boldsymbol{\lambda}}^k$ (e.g. limited minimization rule, Armijo rule, etc.), and $\Delta\boldsymbol{\theta}^k$, $\Delta\boldsymbol{\lambda}^k$ (e.g. feasible direction, descent direction, steepest direction, etc.), the sequences $\{\boldsymbol{\theta}^k\}$ and $\{\boldsymbol{\lambda}^k\}$ can converge to $\boldsymbol{\theta}^*$ and $\boldsymbol{\lambda}^*$ respectively.*

## 1.3 Two main approaches

In brief, the iterative and analytical methods are two main approaches to solve MPC resultant optimization problem (1.5).

The iterative method was first studied in 1956 by Frank and Wolfe [20], and has been advanced over the last decades in combination with dual decomposition and methods for general nonlinear and convex optimization [41], which mainly lies in 2 classes: the first order method and the second order method.

The first order method often possesses a linear convergence rate, in which the gradient or sub-gradient of the objective and constraint functions is used to form the search direction, including gradient method [2], sub-gradient method [6], conjugate gradient method [57], etc. By virtue of simplicity implementation, insensitivity to dimension change, and light burden in computing the search direction, the first order method is desirable for iterative process (1.9), especially for distributed structure.

The second order method, where the second order gradient of objective and constraint functions is used to form the search direction, is entitled to much faster convergence rate (usually quadratic) but more complicated implementation, e.g. Newton method [29], quasi Newton method [6], Gauss Newton method [64], etc. Its formidable rapid convergence,

usually several tens of iterations [8] can reach extremely high accuracy, granting it a high priority to be employed to all kinds of convex optimization problems.

The analytical methods to solve problem (1.5) with (1.5c) being linear can be classified into two categories: the numerical and geometrical methods.

The numerical method, using multi-parametric quadratic programming (mp-QP) [1], was initially proposed in [5], where the initial state was deemed as the multi-variables to form an offline mapping by partitioning its Euclidean space into neighboring critical regions. That is obtained via two steps: first, solve a Karush-Kuhn-Tucker (KKT) condition based linear programming for a polyhedron with a given feasible starting point; second, visit the opposite side of the polyhedron border (hyperplane) one by one to form other critical regions.

A great deal of research, related to analytical numerical method, has been developed either to extend its application scope or to improve its efficiency, including reduction of unnecessary critical region partition [61], non linear independence constrains qualification cases and semi-definite hessian cases [60], optimal active set transfer in prediction horizon $N-1$ to $N$ [37] [36], pruning infeasible set during critical region partition [26], using graph traversal technique with no nondegeneracy requirement [46]. It is worth noting that in practice, entirely visiting all regions is hugely time-consuming and memory exhausting even for medium size problem[33], and the "ergodic" property of critical region visit limits its applicability only for small size system [63].

The geometrical analytical method, which was proposed in [56], takes advantage of the geometry property of QP to construct an ellipsoid centered at the unconstrained optimal solution, by which the desired solution is found at the nearest point in each region partitioned by hyperplane and its normals. However, this method is limited to box constraints cases.

From §2 to §5, several distinct methods, including the first order method, and the second order method, will be applied to solve problem (1.5), depending on different configurations and assumptions. A more detailed literature review will be given at the beginning of each chapter.

## 1.4 Motivation and main contributions

### 1.4.1 Motivation

A universal but pessimistic fact about almost all kinds of iterative methods is that the optimality and feasibility, in terms of primal problem (1.5), are only guaranteed in the limit of iterations when (1.9) is applied [8]. Consequently, from both arithmetical (the iterative process must be terminated within finite iterations; otherwise no solution is attained) and practical (all simulations and applications prefer the solving time to be as short as possible) perspective, a stopping condition is compulsory to enforce in implementing (1.9).

Since the solution obtained by such a condition is not ensured to be optimal, the motivation of this dissertation is to accelerate the termination of the iterative process (1.9) and deliver solutions that can satisfy the predetermined criteria, which is realized by exploitation of the temporal structure of problem (1.5) showed in (1.6) and the MPC feature that only the first step of control sequence is applied to the system.

### 1.4.2 Main contributions

The main contributions of this dissertation are listed as follows:

1. new treatments to accelerate termination of the iterative process, including 2 algorithms proposed in §2 to reduce the complexity and enhance the efficiency during iterations;

2. new stopping conditions to guarantee suboptimality, including gradient based stopping condition proposed in §3, and active set identification based stopping condition proposed in §5, where the mathematical proof of suboptimality lower bound is given;

3. new treatments to ensure the feasibility, including projection mechanism proposed in §3, cone programming based approach proposed in §5;

4. the improved stopping conditions with suboptimality and feasibility guarantee for MPC (the MPC feature that only the first step components of the control sequence are applied to the system is exploited), including gradient based stopping condition proposed in §3, and modified KKT condition based criterion proposed in §4.

## 1.5 Outline

After presenting the general introduction of the dissertation in §1, the techniques of reducing the iterations requirement in a distributed MPC are illustrated in §2, where the bound of suboptimality is demonstrated for two newly proposed algorithms. In §3, gradient based suboptimality stopping conditions are constructed for the full prediction horizon, and a first step focused projection is proposed to produce feasible and suboptimality guaranteed solution. In §4, the first step focused suboptimality stopping condition based on the primal dual interior point method is demonstrated for MPC problem with general convex inequality constraints. §5, taking a step out of the MPC territory, delineates the situation for general linear constrained quadratic optimization. In §5, a suboptimality condition incorporating cone programming is designed to generate feasible solutions with suboptimality guarantee, enabling faster termination of the iterative process. A road map is depicted in Fig. 1.1, where an overall picture of each chapter's advancement route and features is given.

In this dissertation, as each chapter tackles with a specific form of problem (1.5) (characterized by more simplified or more generalized (1.5b) or (1.5c)) by employing different methods (the first order or second order method, centralized or distributed structure), the brief introduction of each chapter is presented here, including the problem differentiation, the know-how of the technique proposed and exact contributions. As such, the reader can catch a glimpse of the essence of each chapter without digging deep into details.

### §2: dynamic reduction of the iterations requirement in a distributed MPC

In §2, a special case of problem (1.5) is dealt with, where no interaction of dynamics among subsystems is considered, say the dynamics is decoupled. §2 is mainly based on publication [17]. In addition, inequality constraints occurred in §2 are specialized form of (1.5c), which consists of local and global inequality constraints. The latter moreover is considered to be separated into components of each subsystems.

§2 starts with an initiative to reduce the iteration number in solving the problem resulted from distributed MPC. To this aim, dynamic Lagrange multipliers fixation algorithm (DLMFA) is proposed by continually fixing the value of Lagrange multipliers, and local optimization problems dynamic sizing algorithm (LOPDSA) is proposed by continually reducing the size of local optimization problem during the iterations through

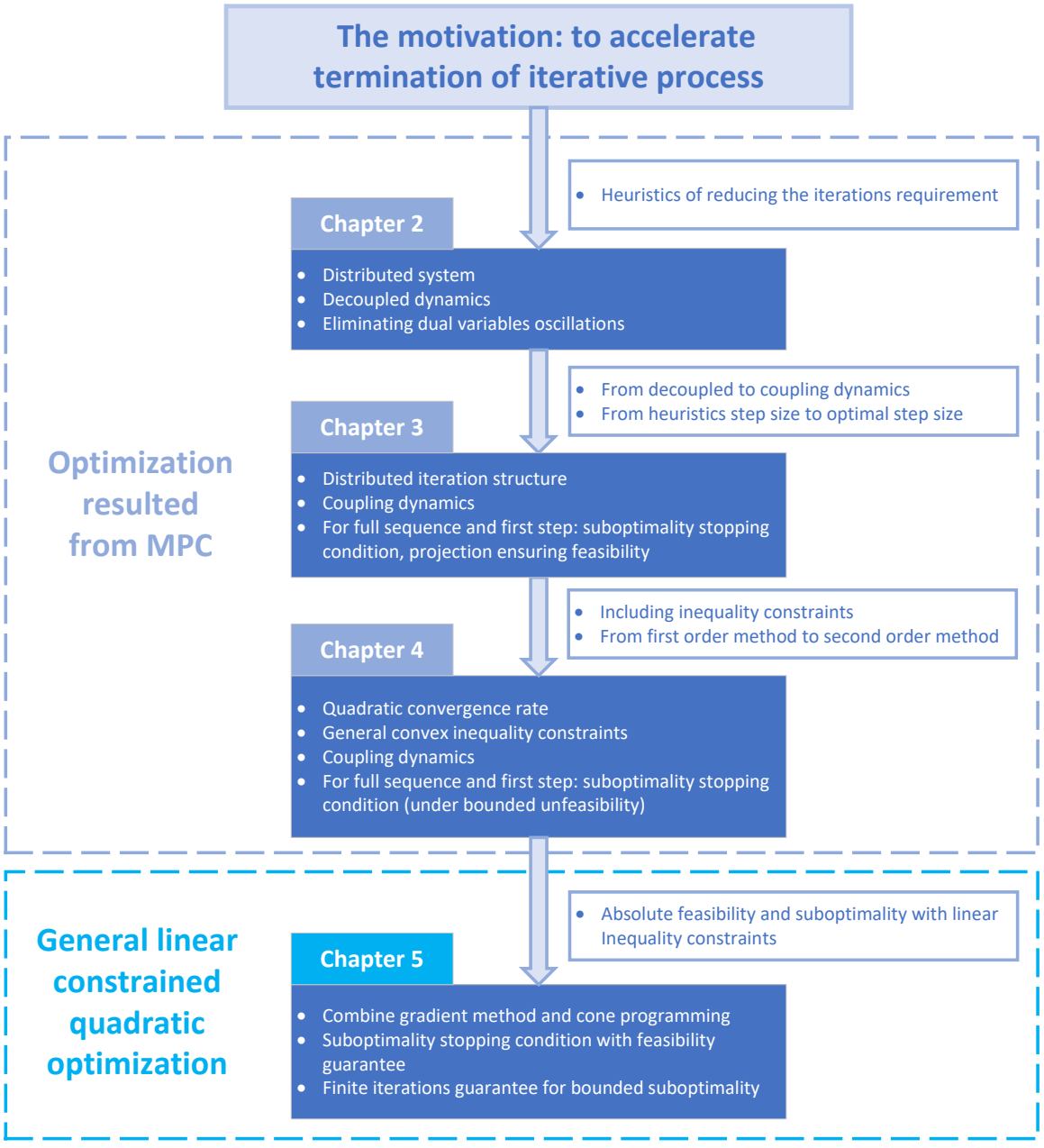


Figure 1.1 – Structure and technique road map of the dissertation
The downward arrows stand for evolution direction from the chapter above (or general heuristics) to the chapter below. The box at the right of each arrow means the main improvement or generalization of the chapter below compared to the chapter above. The blue-filled box under each chapter indicates the main features of the problem or method that appeared therein.

an original prediction horizon reduction. The proposed algorithms improve the Uzawa method's performance by exploiting the step separable constraints in the MPC context. These improvements stem from the particular behavior of the Lagrange multipliers and their fluctuations over the prediction horizon. Numerical experiments show that the iteration number and the computation time of LOPDSA are significantly reduced compared to the Uzawa method.

## §3: $\epsilon$ suboptimality based accelerated termination for equality constrained MPC

In §3, the equality constrained case of problem (1.5) is considered.

Dual decomposition is an efficient tool in dealing with MPC problems, particularly for distributed MPC. In §3, limiting the iteration number is proposed by stopping the iterative process once the solution is close enough to the optimal one. The concepts of primal and dual suboptimality are introduced, and projection mechanisms and stopping conditions are derived, respectively. By exploiting the particular structure of the MPC problem where only the first step inputs are applied to the system, a faster $\epsilon$ suboptimality stopping condition is devised. Focusing on components only at the first step of the prediction horizon, the proposed algorithm can further reduce the iteration number needed. Beyond the theoretical proofs developed, the method's efficiency, both in computation time and iteration number, is illustrated by various simulations.

## §4: $\epsilon$ suboptimality based accelerated termination for MPC using primal dual interior point method

In §4, a step separable structure of the inequality constraints (1.3) is considered.

Employing primal dual interior point method to MPC generated problem to solve for a suboptimal solution is a mature approach with satisfying performance. In §4, the step-based structure of dynamics and inequality constraints of the system is exploited. A first step focused stopping criterion with the guarantee of predefined suboptimality is proposed. The method that integrated this newly criterion is superior on iteration number than the existing primal dual interior point method. In addition to the mathematical proofs provided, various simulations illustrate the effectiveness and efficiency of the method.

## §5: $\epsilon$ suboptimality based accelerated termination for linear constrained quadratic optimization

In §5, a more generalized form of (1.5b) is considered, where no dynamics of states or inputs are specified, but the general linear equality constraints are imposed to variables. In particular, for inequality constraints (1.3), the general unstructured linear form is considered.

Many methods use dual decomposition to solve linear constrained quadratic optimization problems. An iterative mechanism ensures the convergence towards the optimal solution in all these methods. However, the convergence is only guaranteed in the limit of iterations. In §5, a degree of suboptimality is introduced to terminate the iterative process faster while ensuring the fulfillment of constraints. One of the primary keys of this work is identifying the active inequalities constraints during the iterative process. In addition to the mathematical proofs provided, various simulations illustrate the effectiveness of the method proposed.

Chapter 2

# Dynamic Reduction of The Iterations Requirement in A Distributed MPC

In this chapter, a distributed system structure, consisting of decoupled dynamics and separable global inequality constraints, is considered. The resulting MPC problem is first solved by the Uzawa method, of which the persistent oscillation of Lagrange multipliers during the iterations is then investigated. Two algorithms, aiming at eliminating the fluctuate behaviors of Lagrange multipliers, are proposed to terminate the iterative process faster.

## 2.1 Introduction

Computational techniques and algorithms have been widely studied to improve the Uzawa method's performance or extend its applicability. Among all these works, a fundamental one is the Augmented Lagrangian (AL) method[25], for which an additional quadratic penalty is introduced to tackle the relaxed constraints. By doing so, the dual problem iteration is robustified. The Alternating Direction Method of Multipliers (ADMM) is derived by applying the quadratic penalty in a distributed context, which has gained popularity in recent years due to its better feasibility and rapid convergence[7][62]. In various circumstances, it also has been studied for limited communication among network nodes[38] and hierarchical structure[9].

Other researches, focused on the decomposition method and gradient, have been carried out as well for distributed MPC. As in layering decomposition, nodes of a network are partitioned to form several layers, each of which is in charge of an optimization subproblem and thus coordinates with interface variables[13]. Combined with dual decomposition and accelerated gradient method, the algorithm proposed in [24] obtains a much faster

convergence rate compared to regular gradient method.

In this chapter, the goal is to reduce the iteration number required in iterative process based on the particular structure of the MPC problem. In detail, it can be noticed that there is a precise sequencing in the convergence of Lagrange multipliers, which is then exploited to propose a continuous reduction in the complexity of local problems during the iterative process. The proposed strategy achieves a bounded suboptimal solution with a significant reduction of computation effort compared to the Uzawa method.

This chapter is organized as follows. In §2.2, the system and resulting optimization problem are formulated. §2.3 presents the behavior of the Lagrange multipliers during the Uzawa method. §2.4 proposes DLMFA and LOPDSA to reduce the number of iterations and study their suboptimality. Experiments and results are presented in §2.5 and conclusions are given in §2.6.

## 2.2 Problem statement

In this section, the dynamics and constraints of system considered are presented first, the resulting MPC problem is formulated subsequently.

### 2.2.1 Centralized problem statement

The decoupled dynamics of the system is:

$$x_l(j) = A_{ll}x_l(j-1) + B_{ll}u_l(j-1),\ l = 1, ..., m,\ j = 1, ..., N, \tag{2.1}$$

where $x_l(j) \in \mathbb{R}^{nx_l}$ and $u_l(j-1) \in \mathbb{R}^{nu_l}$ are states at step $j$ and inputs at step $j-1$ of $l$-th subsystem, $A_{ll} \in \mathbb{R}^{nx_l \times nx_l}$, $B_{ll} \in \mathbb{R}^{nx_l \times nu_l}$ are system matrix, and $x_l(0) = \bar{x}_l$, $\bar{x}_l \in \mathbb{R}^{nx_l}$ is the initial state of $l$-th subsystem, and denote $\boldsymbol{x}_0 = (x_1(0)^T, ..., x_m(0)^T)^T$.

The local and global inequality constraints imposed on states and inputs are respectively as: for $\forall\ j = 1, ..., N$, $l = 1, ..., m$,

$$f_j^l(x_l(j), u_l(j-1)) \leq \mathbf{0}, \tag{2.2}$$

$$\sum_{l=1}^{m} \hat{f}_l(x_l(j), u_l(j-1)) \leq \mathbf{0}, \tag{2.3}$$

where $f_j^l : \mathbb{R}^{nx_l} \times \mathbb{R}^{nu_l} \to \mathbb{R}^{n_l}$, $\hat{f}_l : \mathbb{R}^{nx_l} \times \mathbb{R}^{nu_l} \to \mathbb{R}^{n_g}$, and for the consistency of total

inequality constraints number defined in (1.3), it holds that $N(mn_l + n_g) = n$.

With decoupled dynamics (2.1), local inequality constraints (2.2), and global inequality constraints (2.3), the resulting centralized problem is formulated as below.

$$J^* = \min_{\boldsymbol{u},\boldsymbol{x}} \ J(\boldsymbol{u}, \boldsymbol{x}), \tag{2.4}$$
$$s.t. \ (2.1), (2.2), (2.3).$$

Here, objective function $J(\boldsymbol{u}, \boldsymbol{x})$ can be decomposed into local parts as:

$$J(\boldsymbol{u}, \boldsymbol{x}) = \sum_{l=1}^{m} \mathcal{J}_l(U_l, X_l),$$
$$\mathcal{J}_l(U_l, X_l) = \sum_{j=1}^{N} \frac{1}{2}(||u_l(j-1)||^2_{R^u_{lj}} + ||x_l(j)||^2_{R^x_{lj}}),$$

where $U_l = (u_l(0)^T, ..., u_l(N-1)^T)^T$, and $X_l = (x_l(1)^T, ..., x_l(N)^T)^T$.

### 2.2.2 Dual decomposition method and distributed structure

Due to global constraints (2.3), problem (2.4) cannot be solved in a distributed manner directly. Hence, the Lagrange multipliers associated with (2.3) is introduced to form the Lagrangian in a distributed structure as follows:

$$L(\boldsymbol{u}, \boldsymbol{x}, \bar{\boldsymbol{\lambda}}) = \sum_{l=1}^{m} L_l(U_l, X_l, \bar{\boldsymbol{\lambda}}), \tag{2.5}$$
$$L_l(U_l, X_l, \bar{\boldsymbol{\lambda}}) = \mathcal{J}_l(U_l, X_l) + \sum_{j=1}^{N} \bar{\boldsymbol{\lambda}}_j^T \hat{f}_l(x_l(j), u_l(j-1)), \tag{2.6}$$

where $\bar{\boldsymbol{\lambda}} = (\bar{\boldsymbol{\lambda}}_1^T, ..., \bar{\boldsymbol{\lambda}}_N^T)^T$, and $\bar{\boldsymbol{\lambda}}_j \in \mathbb{R}^{n_g}_{\geq 0}$ is the Lagrange multiplier associated with constraints (2.3) at step $j$.

The Lagrangian dual functions can be introduced under constraints (2.1) (2.2):

$$\mathcal{G}(\bar{\boldsymbol{\lambda}}) = \sum_{l=1}^{m} \mathcal{G}_l(\bar{\boldsymbol{\lambda}}), \tag{2.7}$$
$$\mathcal{G}_l(\bar{\boldsymbol{\lambda}}) = \min_{U_l, X_l} L_l(U_l, X_l, \bar{\boldsymbol{\lambda}}), \tag{2.8}$$
$$s.t. \ (2.1), (2.2).$$

With abovementioned dual functions, the dual problem of the centralized MPC problem can be defined as:

$$\mathcal{G}^* = \max_{\bar{\boldsymbol{\lambda}}} \mathcal{G}(\bar{\boldsymbol{\lambda}}), \tag{2.9a}$$

$$s.t.\ \bar{\boldsymbol{\lambda}} \geq \mathbf{0}. \tag{2.9b}$$

Fit Remark 1 into the notation of this chapter, it gives:

$$\mathcal{G}^* = J^*. \tag{2.10}$$

**Remark 3.** *Based on Assumption 2 and the fact that constraints* (2.1) *and* (2.2) *of problem 2.4 are completely local for each subsystem, given any feasible* $\bar{\boldsymbol{\lambda}}$ *the local dual function* $\mathcal{G}_l(\bar{\boldsymbol{\lambda}})$ *can be consequently solved as independent convex minimization of* $L_l(U_l, X_l, \bar{\boldsymbol{\lambda}})$ *with corresponding solution satisfying* (2.1) *and* (2.2)*, for which there are abundant mature methods[6][8] and solvers (MOSEK, CPLEX, etc.) to address. As a result, constraints*(2.1) *and* (2.2) *are exempted from formulating Lagrangian* (2.5) (2.6)*, which on one hand helps to maintain a concise notation of this chapter, on the other hand would not degenerate the performance and demonstration of aftermentioned algorithms.*

## 2.3 Uzawa algorithm and its behavior

### 2.3.1 Uzawa algorithm

At the start of $(k+1)$-th iteration, $(\boldsymbol{u}^k, \boldsymbol{x}^k, \bar{\boldsymbol{\lambda}}^k)$ is known from $k$-th iteration. The Lagrange multipliers can be iterated along their subgradient direction in search of the maximum of Lagrangian dual function as:

$$\bar{\boldsymbol{\lambda}}^{k+1} = \bar{\boldsymbol{\lambda}}^k + \alpha^k \hat{\boldsymbol{f}}(\boldsymbol{x}^k, \boldsymbol{u}^k), \tag{2.11}$$

where $\alpha^k$ is the step size in $(k+1)$-th iteration, and

$$\hat{\boldsymbol{f}}(\boldsymbol{x}^k, \boldsymbol{u}^k) = ((\sum_{l=1}^{m} \hat{f}_l(x_l^k(1), u_l^k(0)))^T, ..., (\sum_{l=1}^{m} \hat{f}_l(x_l^k(N), u_l^k(N-1)))^T)^T.$$

Then, for $l$-th subsystem:

$$\mathcal{G}_l(\bar{\boldsymbol{\lambda}}^{k+1}) = \min_{U_l, X_l} L_l(U_l, X_l, \bar{\boldsymbol{\lambda}}^{k+1}), \quad s.t.\ (2.1), (2.2). \tag{2.12}$$

By solving (2.12), $U_l^{k+1}$ is obtained for each subsystem.

Generally, the stopping conditions $\mathcal{C}$ of Uzawa algorithm could be formulated in the form as:

$$\mathcal{C} = (\mathcal{C}_1^T, ..., \mathcal{C}_N^T)^T, \tag{2.13}$$

And for $j = 1, ..., N$, the $\mathcal{C}_j$ could be typically defined as the logical conjunction of these 3 inequality below:

$$||\bar{\boldsymbol{\lambda}}_j^{k+1} - \bar{\boldsymbol{\lambda}}_j^k||_\infty \leq \epsilon_\lambda, \tag{2.14}$$

$$\sum_{l=1}^{m} \hat{f}_l(x_l(j), u_l(j-1)) \leq \epsilon_f \cdot \mathbf{1}, \tag{2.15}$$

$$||\boldsymbol{u}_j^{k+1} - \boldsymbol{u}_j^k||_\infty \leq \epsilon_u, \tag{2.16}$$

where $\epsilon_\lambda \in \mathbb{R}_{>0}$, $\epsilon_h \in \mathbb{R}_{>0}$ and $\epsilon_u \in \mathbb{R}_{>0}$ are respectively thresholds of dual variable convergence, global constraints fulfillment and primal variable convergence.

**Algorithm 1** Uzawa Algorithm

1: Initialize $\boldsymbol{x}_0$, $\boldsymbol{x}^0$, $\boldsymbol{u}^0$ and $\bar{\boldsymbol{\lambda}}^0$.
2: Set $k = 0$
3: **while** $\mathcal{C}$ is not satisfied **do**
4: Update $\bar{\boldsymbol{\lambda}}^{k+1}$ by (2.11)
5: **solve** $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$, $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$ by $\boldsymbol{x}_0$, $\bar{\boldsymbol{\lambda}}^{k+1}$
6: $k \leftarrow k + 1$
7: **end while**
8: **return** $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$, $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$

### 2.3.2 Behavior of the Uzawa algorithm during the iteration

Applying the Uzawa algorithm, it can be constantly observed that the Lagrange multipliers of the first few steps in the prediction horizon always converge faster than the

later steps, whose convergence is delayed by visible fluctuations as illustrated in Fig. 2.1. The fact is that the convergence of Lagrange multipliers is sequential, based on which more efficient algorithms can be proposed.

First, the compact expression of local objective function according to problem (2.4) is formulated as:

$$\bar{J}_l(U_l) = ||X_l||^2_{\boldsymbol{R}_l^x} + ||U_l||^2_{\boldsymbol{R}_l^u}, \tag{2.17}$$

$$X_l = F_l x_l(0) + E_l U_l, \tag{2.18}$$

where $\boldsymbol{R}_l^x \in \mathbb{S}_+^{Nnx_l}$, $\boldsymbol{R}_l^x = \text{blkdiag}(R_{l1}^x, ..., R_{lN}^x)$, $\boldsymbol{R}_l^u \in \mathbb{S}_{++}^{Nnu_l}$, $\boldsymbol{R}_l^u = \text{blkdiag}(R_{l1}^u, ..., R_{lN}^u)$ , $F_l \in \mathbb{R}^{Nnx_l \times nx_l}$, and $F_l = (A_{ll}^T, (A_{ll}^2)^T, ..., (A_{ll}^N)^T)^T$, $E_l \in \mathbb{R}^{Nnx_l \times Nnu_l}$, and

$$E_l = \begin{bmatrix} B_{ll} & 0 & \dots & 0 \\ A_{ll}B_{ll} & B_{ll} & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A_{ll}^{N-1}B_{ll} & \cdots & A_{ll}B_{ll} & B_{ll} \end{bmatrix},$$

In accordance, (2.18) can be rewritten as:

$$\begin{bmatrix} x_l(1) \\ x_l(2) \\ \vdots \\ x_l(N) \end{bmatrix} = \begin{bmatrix} A_{ll} \\ A_{ll}^2 \\ \vdots \\ A_{ll}^N \end{bmatrix} x_l(0) + \begin{bmatrix} Bu_l(0) \\ \sum_{j=0}^{1} A_{ll}^{1-i} B_{ll} u_l(j) \\ \vdots \\ \sum_{j=0}^{N-1} A_{ll}^{N-1-i} B_{ll} u_l(j) \end{bmatrix}.$$

Among the input variables of all time steps, $u_l(0)$ has clearly the highest impact on (2.17) based on the lower triangular structure of $E_{l,N}$ in (2.18).

Next, introduce $\bar{L}(\boldsymbol{u}, \bar{\boldsymbol{\lambda}}) = \sum_{l=1}^{m} \bar{L}_l(U_l, \bar{\boldsymbol{\lambda}})$, and $\bar{L}_l(U_l, \bar{\boldsymbol{\lambda}}) = \sum_{j=1}^{N} \bar{L}_{l,j}(U_l, \bar{\boldsymbol{\lambda}})$ can be presented with step separable structure as:

$$\begin{bmatrix} \bar{L}_{l,1}(U_l, \bar{\boldsymbol{\lambda}}) \\ \vdots \\ \bar{L}_{l,N}(U_l, \bar{\boldsymbol{\lambda}}) \end{bmatrix} = \begin{bmatrix} \bar{J}_{l,1}(U_l) \\ \vdots \\ \bar{J}_{l,N}(U_l) \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{\lambda}}_1^T \bar{f}_l(u_l(0)) \\ \vdots \\ \bar{\boldsymbol{\lambda}}_N^T \bar{f}_l(u_l(N-1)) \end{bmatrix}, \tag{2.19}$$

where for $l = 1, ..., m$, and $j = 1, ..., N$, by substituting (2.1), $\bar{f}_l(u_l(j-1)) = \hat{f}_l(x_l(j), u_l(j-1)))$, $\bar{f}_l : \mathbb{R}^{nu_l} \to \mathbb{R}^{n_g}$, and denote $\bar{\boldsymbol{f}}_l(U_l) = ((\bar{f}_l(u_l(0)))^T, ..., (\bar{f}_l(u_l(N-1)))^T)^T$.

Given that $u_l(j-1)$ at each step in $\hat{f}_l(x_l(j), u_l(j-1))$ possess the same weight, obviously the input variables of the first time step, $u_l(0)$, account the highest weight in local Lagrangian among input variables of all time steps.

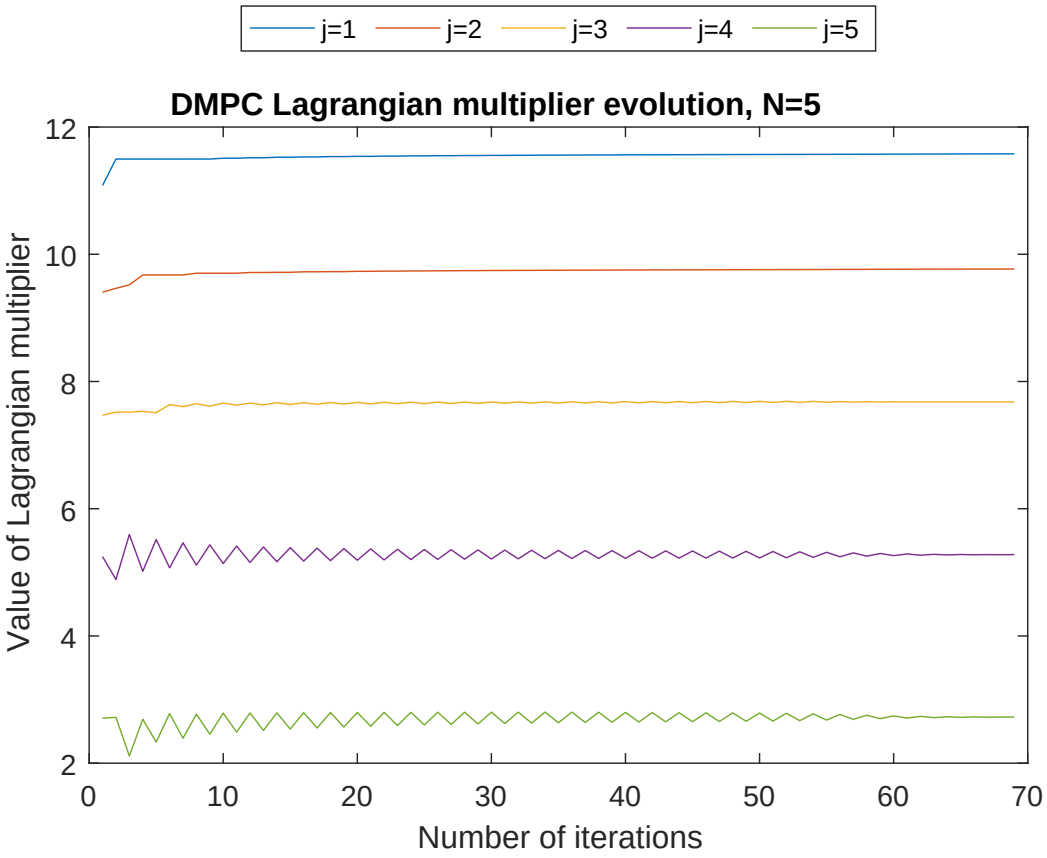


Figure 2.1 – Typical Lagrange multiplier evolution with Uzawa algorithm with $N = 5$

## 2.4 Dynamic reduction of the iterations requirement in a distributed MPC

Based on the analysis and conclusions drawn in §2.3, there is an incentive to eliminate the fluctuations of Lagrange multipliers in iterations and accelerate the convergence by features of distributed MPC.

### 2.4.1 Dynamic Lagrange multipliers fixation algorithm

As a start, the idea of fixing the value of Lagrange multipliers of earlier satisfied steps along the prediction horizon is proposed to remove the fluctuations of steps already satisfied, which can mitigate the possible disturbance caused by these fluctuations to later steps' iteration.

DLMFA is introduced hereinafter by the following rules:

1. the fixation should always start from the first step in an active sequence (an active sequence is the remaining part of the prediction horizon whose first some steps are fixed before);

2. the fixation should proceed continually from the first to the last step in an active sequence.

**Algorithm 2** Dynamic Lagrange Multipliers Fixation Algorithm (DLMFA)

Initialize $\boldsymbol{x}_0$, $\boldsymbol{x}^0$, $\boldsymbol{u}^0$ and $\bar{\boldsymbol{\lambda}}^0$.
Set $k = 0$ and $d_t = 0$.
▷ $d_t$ is the number of steps of Lagrange multipliers has been fixed before the current iteration
**while** $\mathcal{C}_{(d_t+1:N)}$ is not satisfied **do**
    **if** $d_t > 0$ **then**
        $\bar{\boldsymbol{\lambda}}^{k+1}_{(1:d_t)} \leftarrow \bar{\boldsymbol{\lambda}}^{k}_{(1:d_t)}$
    **end if**
    $\bar{\boldsymbol{\lambda}}^{k+1}_{(d_t+1:N)} \leftarrow \bar{\boldsymbol{\lambda}}^{k}_{(d_t+1:N)} + \alpha^k \hat{\boldsymbol{f}}_{(d_t+1:N)}(\boldsymbol{x}^k, \boldsymbol{u}^k)$
    **solve** $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$, $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$ by $\boldsymbol{x}_0$, $\bar{\boldsymbol{\lambda}}^{k+1}$
    **if** $\mathcal{C}_{(d_t+1:d_t+d)}$ is satisfied and $d > 0$ **then**
        $d_t \leftarrow d_t + d$ ▷ update $d_t$ when new fixation of Lagrange multipliers happens
    **end if**
    $k \leftarrow k + 1$
**end while**
**return** $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$, $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$

Nevertheless, facts should be noticed that supposing $\bar{\boldsymbol{\lambda}}_j$ is fixed, while $\boldsymbol{u}_j$ is still served as variables in solving $\mathcal{G}_l(\bar{\boldsymbol{\lambda}})$ and its value will definitely be changed due to later iterations, making the conditions (2.15)(2.16) might be failed in the final solution. As a result, this may cause worse suboptimality. Consequently, a better proposal is to "fix" $\boldsymbol{u}_j$ as well once $\bar{\boldsymbol{\lambda}}_j$ is fixed during the iteration, which can be achieved by dynamically redefining the local optimization problem's size and will be demonstrated in §2.4.2.

### 2.4.2 Local optimization problems dynamic sizing algorithm

**Algorithm 3** Local Optimization Problems Dynamic Sizing Algorithm (LOPDSA)

Initialize $\boldsymbol{x}_0$, $\boldsymbol{x}^0$, $\boldsymbol{u}^0$ and $\bar{\boldsymbol{\lambda}}^0$.
Set $k = 0$ and $d_t = 0$. ▷ $d_t$ is the number of steps has been dropped before the current iteration
**while** $\mathcal{C}_{(d_t+1:N)}$ is not satisfied **do**
    $\bar{\boldsymbol{\lambda}}^{k+1}_{(d_t+1:N)} \leftarrow \bar{\boldsymbol{\lambda}}^{k}_{(d_t+1:N)} + \alpha^k \hat{\boldsymbol{f}}_{(d_t+1:N)}(\boldsymbol{x}^k_{(d_t+1:N)}, \boldsymbol{u}^k_{(d_t+1:N)})$
    **solve** $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1}_{(d_t+1:N)})$, $\boldsymbol{x}^{k+1}_{(d_t+1:N)}$ and $\boldsymbol{u}^{k+1}_{(d_t+1:N)}$ by $\boldsymbol{x}_{d_t}$, $\bar{\boldsymbol{\lambda}}^{k+1}$
    **if** $\mathcal{C}_{(d_t+1:d_t+d)}$ is satisfied and $d > 0$ **then**
        save results: $\hat{\boldsymbol{u}}_{(d_t+1:d_t+d)} \leftarrow \boldsymbol{u}^{k+1}_{(d_t+1:d_t+d)}$
        **get** $x_l(d_t + d)$ by (2.1), then get $\boldsymbol{x}_{d_t+d}$
        substitute $\boldsymbol{u}^{k+1}_{(d_t+d:N-1)}$ into $\nabla \boldsymbol{u}_{(d_t+d:N-1)} \bar{L}_{(d_t+d+1:N)}(\boldsymbol{u}_{(d_t+d:N-1)}, \bar{\boldsymbol{\lambda}}_{(d_t+d+1:N)}) = \mathbf{0}$, **to rebuild** $\bar{\boldsymbol{\lambda}}^{k+1}_{(d_t+d+1:N)}$
        $d_t \leftarrow d_t + d$ ▷ update $d_t$ when new drop happens
    **end if**
    $k \leftarrow k + 1$
**end while**
substitute $\hat{\boldsymbol{u}}$ into $\nabla_{\boldsymbol{u}} \bar{L}(\boldsymbol{u}, \bar{\boldsymbol{\lambda}}) = \mathbf{0}$, **to rebuild** $\hat{\boldsymbol{\lambda}}$
**get** $\bar{L}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}})$ by $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{\lambda}}$ and $\boldsymbol{x}_0$
$\mathcal{G}(\hat{\boldsymbol{\lambda}}) \leftarrow \bar{L}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}})$
**return** $\mathcal{G}(\hat{\boldsymbol{\lambda}})$, $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$

To dynamically fix $\boldsymbol{u}(j)$ and redefine the local optimization problem, the step $j$ from the solving process of $\mathcal{G}(\bar{\boldsymbol{\lambda}})$ can be removed when $\mathcal{C}_j$ is satisfied, and by updating the initial state with fixed $\boldsymbol{u}(j)$, the solution obtained would finally coincide with problem (2.7).

Without loss of generality, suppose that after $k$-th iteration, $d$ ($d \in \mathbb{N}_{>0}$) steps are to be dropped and no step has been dropped before. Then the new local problem after drop for $l$-th subsystem is:

$$\min_{U_{l,(d:N-1)}} \quad ||X_{l,(d+1:N)}||^2_{\boldsymbol{R}^x_{l,(d+1:N)}} + ||U_{l,(d:N-1)}||^2_{\boldsymbol{R}^u_{l,(d+1:N)}}, \tag{2.20a}$$

$$s.t.\ X_{l,(d+1:N)} = F_{l,(1:N-d)} x^k_l(d) + E_{l,(1:N-d)} U_{l,(d:N-1)}, \tag{2.20b}$$

$$f^l_j(x_l(j), u_l(j-1)) \leq \mathbf{0},\ j = d+1, ..., N, \tag{2.20c}$$

$$\sum_{l=1}^{m} \hat{f}_l(x_l(j), u_l(j-1)) \leq \mathbf{0},\ j = d+1, ..., N. \tag{2.20d}$$

Note that (2.20b) is equivalent to (2.18) taking $x_l^k(d)$ as $x_l(0)$ from step $d+1$ to $N$, resulting $F_{l,(1:N-d)}$ and $E_{l,(1:N-d)}U_{l,(d:N-1)}$ presented in the expression.

Recall from MPC: the computation effort increases rapidly with the prediction horizon length, sometimes exponentially, which also adds the impulse to dynamically redefine the size of the local optimization problem, as the prediction horizon is indeed dynamically reduced while solving $\mathcal{G}(\bar{\boldsymbol{\lambda}})$.

It is essential to rebuild the Lagrange multipliers by intermediate input variables of the remaining steps in the prediction horizon whenever the local optimization problem size is redefined. In this way, a "warm start" can be achieved to continue the Lagrange multipliers iteration efficiently.

Recall that $nu = nu_1 + ... + nu_m$, and $n_g$ is the number of global inequality constraints at each time step. The following theorem will demonstrate the condition to rebuild Lagrange multipliers.

**Theorem 1.** *In LOPDSA, the sufficient and necessary condition to completely rebuild Lagrange multipliers under Assumption 2 is:*

$$n_g \leq nu. \tag{2.21}$$

*Proof. Sufficient Condition:* Without loss of generality, assume that after $k$-th iteration $d$ ($d \in \mathbb{N}_{>0}$) steps are to be dropped and no step have been dropped before. As $\hat{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u})$ is differentiable, the necessary condition for newly generated local problem with updated active sequence to achieve the minimum by KKT conditions[8] is as: $\forall l = 1, ..., m$,

$$\nabla_{U_{l,(d:N-1)}} \bar{L}_{l,(d+1:N)}(U_{l,(d:N-1)}, \bar{\boldsymbol{\lambda}}_{(d+1:N)}) = \mathbf{0}. \tag{2.22}$$

Combining (2.22) for $m$ subsystems, a equation group can be formed by (2.18) as below:

$$\begin{cases} E_{1,(1:N-d)}^T \boldsymbol{R}_{1,(d+1:N)}^x X_{1,(d+1:N)} + \boldsymbol{R}_{1,(d+1:N)}^u U_{1,(d:N-1)} + \\ \qquad \nabla \bar{\boldsymbol{f}}_{1,(d:N-1)}(U_{1,(d:N-1)}) \bar{\boldsymbol{\lambda}}_{(d+1:N)} = \mathbf{0}, \\ \qquad\qquad \vdots \\ E_{m,(1:N-d)}^T \boldsymbol{R}_{m,(d+1:N)}^x X_{m,(d+1:N)} + \boldsymbol{R}_{m,(d+1:N)}^u U_{m,(d:N-1)} + \\ \qquad \nabla \bar{\boldsymbol{f}}_{m,(d:N-1)}(U_{m,(d:N-1)}) \bar{\boldsymbol{\lambda}}_{(d+1:N)} = \mathbf{0}. \end{cases} \tag{2.23}$$

(2.23) can be regarded as a linear equation group with $\bar{\boldsymbol{\lambda}}_{d+1:N}$ serving as $(N-d)n_g$ variables to be solved. The positive definite matrix $\boldsymbol{R}_{l,d+1:N}^u$ makes the augmented matrix

of $l$-th equation in (2.23) to be full row rank. If condition (2.21) is satisfied, the augmented matrix rank of (2.23) is therefore $(N-d)nu$.

To conclude, $\bar{\boldsymbol{\lambda}}_{d+1:N}$ could be solved uniquely by (2.23) if equality in (2.21) holds, or be calculated completely (e.g. select $(N-d)n_g$ row from (2.23) by certain rules to form a linear equation with unique solution, or approximate $\bar{\boldsymbol{\lambda}}_{d+1:N}$ completely by least square method, etc.) if strict inequality in (2.21) holds.

*Necessary Condition:* Here (2.21) is shown as the necessary condition by contradiction. Suppose that Lagrange multipliers in LOPDSA could be completely rebuilt, and $n_g > nu$.

With that said, the augmented matrix rank $(N-d)nu$ is now less than the variable number $(N-d)n_g$ in (2.23), which means there exists infinite solutions of this linear equation group. In other words, $\bar{\boldsymbol{\lambda}}_{d+1:N}$ cannot be solved uniquely or be entirely calculated due to insufficient information. Namely, Lagrange multipliers cannot be completely rebuilt, which is a contradiction to the assumption. □

Assume that the condition (2.21) is satisfied, LOPDSA is introduced hereinafter by the following rules:

1. the drop should always start from the first step in an active sequence, only in this way the validity of dynamics in (2.1) can be ensured;

2. the drop should proceed continually from the first to the last step in a prediction horizon. Otherwise, the initial state cannot be updated if discontinuous drops happen in the iteration.

### 2.4.3 Suboptimality of local optimization problems dynamic sizing algorithm

From here, the theoretical bounds for the Lagrangian dual function are derived. To better illustrate the drop mechanism, denote $\tilde{\boldsymbol{\lambda}}_{0,(1:N)} = \bar{\boldsymbol{\lambda}}^*$, and for $1 \le i+1 \le j \le N$, $\tilde{\boldsymbol{\lambda}}_{i,(i+1:j)}$ is the part of the optimal solution $\tilde{\boldsymbol{\lambda}}_{i,(i+1:N)}$ of $\mathcal{G}_{(i+1:N)}$ when $i$ steps have been dropped before.

Suppose that $\epsilon_\lambda$ is sufficiently small, such that when (2.14) is satisfied for the whole prediction horizon, sequence $\{\bar{\boldsymbol{\lambda}}^k\}$ starts to oscillate in the neighbor of $\bar{\boldsymbol{\lambda}}^*$, thus it gives:

$$||\bar{\boldsymbol{\lambda}}_j^{k+1} - \bar{\boldsymbol{\lambda}}_j^*||_\infty \le \epsilon_\lambda,\ j = 1, ..., N, \tag{2.24}$$

$$||\bar{\boldsymbol{\lambda}}^{k+1} - \tilde{\boldsymbol{\lambda}}_{0,(1:N)}||_\infty \le \epsilon_\lambda. \tag{2.25}$$

Since the dual function (2.7) is the pointwise infimum of a family of affine functions of $\bar{\boldsymbol{\lambda}}$, it is concave[8]. By the property of concave function, Uzawa algorithm and DLMFA have:

$$\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1}) \in [M_{(1:N)}(\epsilon_\lambda), J^*], \tag{2.26}$$

where for $1 \leq i \leq j \leq N$ and $e_{(i:j)} \in \mathbb{R}_{>0}$, $M_{(i:j)}(e_{(i:j)}) = \min\{\mathcal{G}_{(i:j)}(\ \max\{0, \bar{\boldsymbol{\lambda}}^{k+1}_{(i:j)} - e_{(i:j)} \cdot \mathbf{1}\}), \mathcal{G}_{(i:j)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(i:j)} + e_{(i:j)} \cdot \mathbf{1})\}$, $M_{(i:j)}(e_{(i:j)})$ denotes the lower bound of $\mathcal{G}_{(i:j)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(i:j)})$ with $i$ as the steps have been dropped before and $e_{(i:j)}$ as the maximum norm of the difference between $\bar{\boldsymbol{\lambda}}^{k+1}_{(i:j)}$ and $\tilde{\boldsymbol{\lambda}}_{0,(1:N)}$. Note that, the subscript 0 in $\tilde{\boldsymbol{\lambda}}_{0,(1:N)}$ means no drop has happened before solving it, which means it is the part of the optimal solution $\bar{\boldsymbol{\lambda}}^*$ of problem (2.9).

As for LOPDSA, the result is illustrated by example $N = 2$ , and suppose that the first step has been dropped before the second step converges. In this case, the suboptimality of $\mathcal{G}_{(1:1)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(1:1)})$ is the same as that of Uzawa algorithm. And it gives: $\mathcal{G}_{(1:1)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(1:1)}) \in [M_{(1:1)}(\epsilon_\lambda), J^*_{(1:1)}]$.

Every time there are steps dropped, a different optimization problem is actually formed for the remaining steps as the prediction horizon and initial state are different since then. Now, (2.25) for the second step becomes:

$$||\bar{\boldsymbol{\lambda}}^{k+1}_{(2:2)} - \tilde{\boldsymbol{\lambda}}_{1,(2:2)}||_\infty \leq \epsilon_\lambda, \tag{2.27}$$

$$||\tilde{\boldsymbol{\lambda}}_{1,(2:2)} - \tilde{\boldsymbol{\lambda}}_{0,(2:2)}||_\infty = e^*_{(2:2)}, \tag{2.28}$$

where for $1 \leq i+1 \leq j \leq N$, $e^*_{(i:j)}$ denotes the maximum norm of the difference between $\tilde{\boldsymbol{\lambda}}_{i,(i+1:j)}$ and $\tilde{\boldsymbol{\lambda}}_{0,(i+1:j)}$, which can be obtained by substituting corresponding centralized solution into (2.22) with constraint (2.9b).

Note that, if (2.3) is removed in problem (2.4), and the resulting solution satisfies (2.3), then in this case the global constraints (2.3) is inactive and $\tilde{\boldsymbol{\lambda}}_{i,(i+1:N)} = \mathbf{0}$.

Now that $e^*_{(2:2)}$ can be solved, the following relation holds:

$$\begin{aligned} ||\bar{\boldsymbol{\lambda}}^{k+1}_{(2:2)} - \tilde{\boldsymbol{\lambda}}_{0,(2:2)}||_\infty &\leq \epsilon_\lambda + e^*_{(2:2)} = e_{(2:2)}, \\ \mathcal{G}_{(2:2)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(2:2)}) &\in [M_{(2:2)}(e_{(2:2)}), J^*_{(2:2)}]. \end{aligned}$$

In the end, for this example $N = 2$:

$$\mathcal{G}_{(1:2)}(\bar{\boldsymbol{\lambda}}^{k+1}_{(1:2)}) \in [M_{(1:1)}(\epsilon_\lambda) + M_{(2:2)}(e_{(2:2)}), J^*_{(1:2)}].$$

From this example, It can be concluded that every time a drop happens, the lower bound of $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$ for LOPDSA decreases. With that said, the possible lowest bound of $\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1})$ for LOPDSA is achieved under the condition that each time only one step drops for all $N$ steps along the prediction horizon. Finally for LOPDSA, when stopping condition (2.14) for whole prediction horizon is satisfied:

$$\mathcal{G}(\bar{\boldsymbol{\lambda}}^{k+1}) \in [M_{(1:1)}(\epsilon_\lambda) + \sum_{i=2}^{N} M_{(i:i)}(e_{(i:i)}), J^*]. \tag{2.29}$$

## 2.5 Numerical experiment

Table 2.1 – Performance comparison among Uzawa algorithm, DLMFA and LOPDSA

| | N | 5 | 10 | 15 | 20 | 25 |
|---|---|---|---|---|---|---|
| Iteration number | Uzawa | 248 | 404 | 1258 | 3502 | 8442 |
| | DLMFA | 248 | 404 | 1252 | 3491 | 8339 |
| | LOPDSA | 248 | 236 | 510 | 1119 | 2007 |
| Computation time (s) | Uzawa | 6.90 | 11.0 | 37.2 | 112 | 312 |
| | DLMFA | 5.96 | 10.6 | 36.0 | 112 | 310 |
| | LOPDSA | 7.81 | 9.73 | 22.4 | 51.1 | 104 |

In this section, all 3 algorithms are tested in numerical experiment with a system consisting of 4 identical subsystems, each of which has 2 states and 1 input. The main parameters of the test are defined as: $\boldsymbol{x}_0 = [5, 10, 15, 20; 5, 10, 15, 20]$, $\bar{\boldsymbol{\lambda}}^0 = \mathbf{0}$, for each $l = 1, ...m$, and $j = 0, ...N-1$, $A_{ll} = [0.9422, 0.0360; 0.0225, 0.8612]$, $B_{ll} = [0.9707; 0.0117]$, the global constraints are $\sum_{i=1}^{m} u_l(j) \geq 1$; the local constraints are $u_l(j) \in [0, 1]$, the step length applied in the example is diminishing non summable as: $\alpha^k = N/\sqrt{k}$; the stopping conditions parameters are $\epsilon_\lambda = 1 \times 10^{-3}$, $\epsilon_h = 1 \times 10^{-2}$ and $\epsilon_u = 1 \times 10^{-2}$. Specifically, the optimization problems are formulated using YALMIP[31] on MATLAB 2018b, solved using `quadprog`, and tested on a Windows 10 PC with 2.20 GHz Core i7-8750H and 16GB RAM.

Table 2.1 shows that as the prediction horizon continues to grow linearly, LOPDSA reduces the iteration number exponentially compared to the Uzawa algorithm and DLMFA.

Table 2.2 – Suboptimality comparison among Uzawa algorithm, DLMFA and LOPDSA with centralized optimal solution as a benchmark

| N | Suboptimality Lower Bound* of Uzawa & DLMFA | Test Suboptimality ($\times 10^{-4}$) | | |
|---|---|---|---|---|
| | | Uzawa | DLMFA | LOPDSA |
| 5 | -4.18 | -2.71 | -2.71 | -2.71 |
| 10 | -34.5 | -1.58 | -13.7 | -57.0 |
| 15 | -102 | -0.49 | -12.9 | -60.1 |
| 20 | -193 | -0.27 | -178 | -29.1 |
| 25 | -309 | -0.01 | -601 | -18.1 |

* Calculated as $M_{(1:N)}(\epsilon_\lambda)$ by substituting $\epsilon_\lambda = 1 \times 10^{-3}$ into (2.25) (2.26).

Besides, starting from $N = 15$, DLMFA reduces the iteration number compared to the Uzawa algorithm, but merely less than 2% in all cases.

As for computation time, DLMFA is slightly lower than the Uzawa algorithm in all cases due to the limited iteration number reduction and its simplified treatment to Lagrange multipliers iteration. On account of more sophisticated treatments in LOPDSA, the advantage of its computation time is mitigated to some extent. Still, it grows exponentially compared to DLMFA and Uzawa algorithm as the prediction horizon increases.

Table 2.2 shows that suboptimality of DLMFA and LOPDSA are within the theoretical bound. It is worth mentioning, suboptimality of DLMFA tends to increase rapidly when $N$ grows, which links to the argument mentioned in §2.4.1.

Figure 2.2 exhibits that all drops/fixes appear continually and homogeneously except one stagnation at 20, revealing the potential in various predictions horizon size application.

## 2.6 Conclusions

In this chapter, the fluctuated behavior during the iteration of the Lagrange multiplier in the Uzawa method applied to distributed MPC has been studied. DLMFA and LOPDSA have been proposed to reduce the iteration requirement. Numerical experiments have showed that in meeting the same stopping conditions with the Uzawa Algorithm, not only the solution precision of LOPDSA has been generally maintained, but its computation time and iteration number have been reduced significantly, especially in large $N$ cases. By

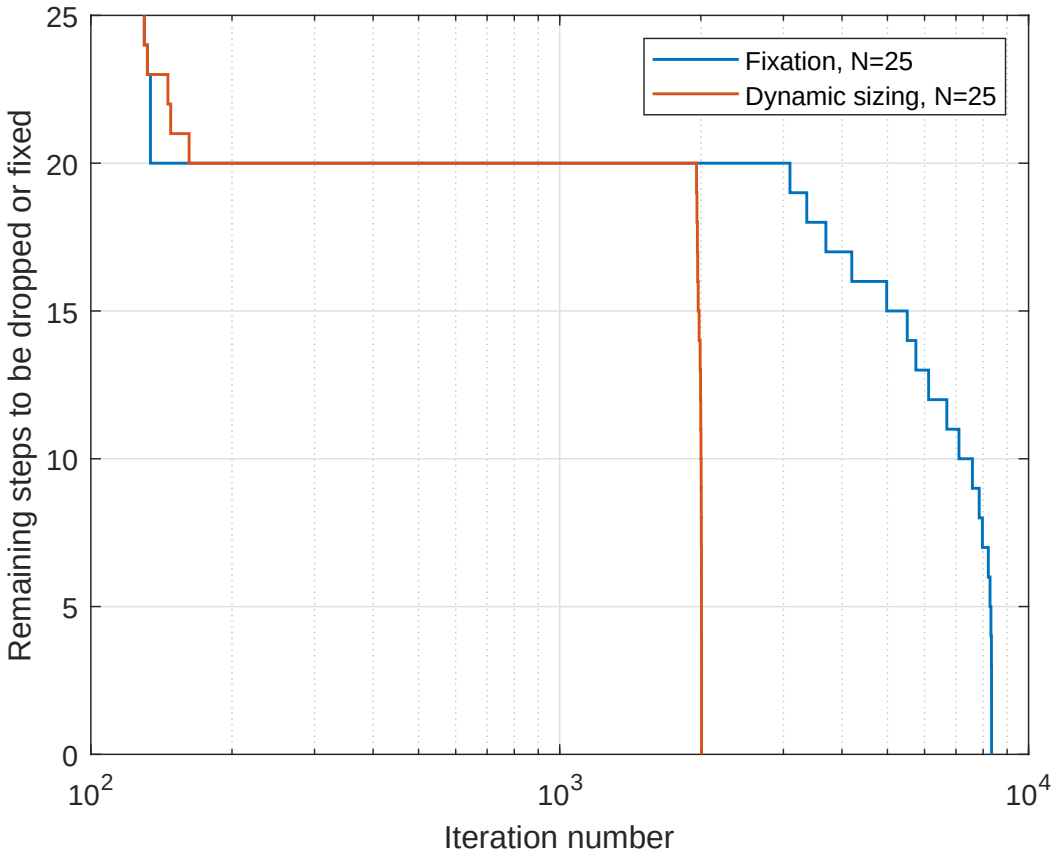


Figure 2.2 – Trajectory of fixes in DLMFA and drops in LOPDSA with $N = 25$

contrast, DLMFA has revealed limited ability to save computation effort and deteriorating suboptimality as the prediction horizon continues to grow.

In §3, in contrast to decoupled dynamics considered in this chapter, the more generalized coupling dynamics among subsystems will be addressed. Furthermore, the decisive parameter in the iterative process, step size, will be updated to the optimal value for the first order method.

Chapter 3

# $\epsilon$ SUBOPTIMALITY BASED ACCELERATED TERMINATION FOR EQUALITY CONSTRAINED MPC

In this chapter, the coupling dynamics among subsystems and global equality constraints are considered. The absence of inequality constraints allows the construction of gradient based criterion, which guarantees the predefined suboptimality and the feasibility ensured projection. The fully distributed iterative process is established by employing the optimal step size of the first order method, the Nesterov gradient descent. The particular characteristic of MPC, only the first step inputs applied to the system, is exploited to derive the first step focused stopping condition and projection mechanism.

## 3.1 Introduction

Due to the attractive accessibility for distributed structure, the updated mainstream techniques in solving optimization problems resulted from MPC are based on the first-order gradient, or subgradient method [69], where the step size is a decisive parameter.

Vast research has been implemented in this field, such as: for fixed step size, e.g., exact first-order algorithm (EXTRA)[58], and distributed inexact gradient method and the gradient tracking technique (DIGing) [39]; for diminishing step size, e.g., distributed (sub)gradient descent (DGD) algorithm [40]. It is worth mentioning that a method originally proposed by Nesterov [42] has been proved by Theorem 2.2.2 of [41] to be the optimal first-order gradient method for strongly convex optimization (with one time continuously differentiable objective function whose first derivative is Lipschitz continuous). The Nesterov gradient descent method has been developed in [3] [42], and has been fitted to MPC in [24].

The updated dual objective value is used to approximate primal optimum in [23] [45] to

generate a suboptimal solution. Still, the lasting approximation gap is presumed to cause over-conservative solutions. Since the dual problem is exactly unconstrained optimization, the gradient-based suboptimality condition can be applied in [8], whose implementation is spared of approximation gap and requires no knowledge of the optimal solution.

Another key problem that arises after the iterative process termination is how to generate a feasible solution with a suboptimality guarantee. One typical way is to employ an adaptable constraint tightening technique during the iteration[23][71], yet heuristic adaption may cause extra iterations. Alternatively, an $\epsilon$ accuracy feasibility, the largest violation of feasible constraints, is introduced in [45] as a stopping condition to obtain a "good enough" solution, which is still not feasible indeed. This chapter proposes a projection mechanism to obtain primal feasible solutions with a suboptimality guarantee based on the gradient norm of dual function.

This chapter is an extension of the work [17], where a heuristic for reducing the local problem size has been elaborated to diminish the complexity in distributed MPC generated optimization. Then main contribution of this chapter is twofold. First, the gradient based stopping condition and projection mechanism can generate feasible solutions with suboptimality guarantee. Second, the first step focused $\epsilon$ suboptimality stopping condition and related projection mechanism can faster the iterative process in generating the first step components of feasible solutions with suboptimality guarantee.

This chapter is organized as follows. §3.2 sets up the optimization problem and fundamentals. §3.3 proposes gradient-based stopping condition and projection ensured feasibility and $\epsilon$-suboptimality. §3.4 demonstrates the first step focused stopping condition and the first step projection with proof of feasibility and $\epsilon$-suboptimality. Numerical experiments and results discussions are presented in §3.5. And conclusions are given in §3.6.

## 3.2 Problem statement and fundamentals

### 3.2.1 Problem statement

To start with, the coupling dynamics and global equality constraints are formed as: for $\forall l = 1, ..., m$, and $j = 1, ..., N$,

$$x_l(j) = \sum_{i=1}^{m} (A_{li} x_i(j-1) + B_{li} u_i(j-1)), \tag{3.1}$$

$$\sum_{l=1}^{m} A_j^l x_l(j) + \sum_{l=1}^{m} B_j^l u_l(j-1) = a_j, \tag{3.2}$$

where $x_l(j) \in \mathbb{R}^{nx_l}$ and $u_l(j-1) \in \mathbb{R}^{nu_l}$ are states at step $j$ and inputs at step $j-1$ of $l$-th subsystem, $A_{li} \in \mathbb{R}^{nx_l \times nx_i}$, $B_{li} \in \mathbb{R}^{nx_l \times nu_i}$ are system matrix, and $x_l(0) = \bar{x}_l$, $\bar{x}_l \in \mathbb{R}^{nx_l}$ is the initial state of $l$-th subsystem, $A_j^l \in \mathbb{R}^{na_j \times nx_l}$, $B_j^l \in \mathbb{R}^{na_j \times nu_l}$ and $a_j \in \mathbb{R}^{na_j}$.

Next, the MPC optimization problem can be formulated in a compact form as:

$$\bar{\mathcal{J}}^* = \min_{\boldsymbol{y}} \ \mathcal{J}(\boldsymbol{y}), \tag{3.3a}$$

$$s.t. \ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \tag{3.3b}$$

where (3.3b) is taken from (1.5b), and please refer to (3.1), (3.2), and (1.5) for composition of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{y}$ respectively.

**Assumption 3.** *In this chapter, $\boldsymbol{R}$ is assumed to be definite positive, and $\boldsymbol{A}$ is assumed to be full row rank.*

Correspondingly, the dual problem of problem (3.3) is formulated as:

$$\bar{g}^* = \max_{\boldsymbol{\theta}} \bar{g}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \min_{\boldsymbol{y}} \bar{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\theta}), \tag{3.4}$$

$$\bar{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{y}||_{\boldsymbol{R}}^2 + \boldsymbol{\theta}^T(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}), \tag{3.5}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$ is the dual variables associated with constraint (3.3b).

### 3.2.2 $\epsilon$-suboptimality definition

**Definition 1.** *$\boldsymbol{y}$ is said to be an $\epsilon$ primal solution of problem (3.3) if and only if: $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$, and $||\mathcal{J}(\boldsymbol{y}) - \bar{\mathcal{J}}^*|| \leq \epsilon$.*

As the dual problem (3.4) is intended to solve prior to get a primal solution, correspondingly the $\epsilon$ suboptimal solution from the dual point of view is defined as follows.

**Definition 2.** *$(\boldsymbol{y}_\theta, \boldsymbol{\theta})$ is said to be an $\epsilon$ dual solution of problem (3.4) if and only if: $\boldsymbol{y}_\theta = \arg\min_{\boldsymbol{y}} \bar{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\theta})$ and $||\bar{\mathcal{L}}(\boldsymbol{y}_\theta, \boldsymbol{\theta}) - \bar{\mathcal{J}}^*|| \leq \epsilon$.*

Note that, an $\epsilon$ suboptimal solution $(\boldsymbol{y}_\theta, \boldsymbol{\theta})$ of problem (3.4) cannot guarantee that $\boldsymbol{y}_\theta$ is primal feasible, which consequently demands a further feasibility verification with respect to problem (3.3).

### 3.2.3 Distributed Nesterov gradient descent method

Here, the Nesterov gradient descent method[24] is adopted for iterative process (1.9), and in context of this chapter it is formulated as:

$$\boldsymbol{\theta}^{k+1} = \hat{\boldsymbol{\theta}}^k + \frac{1}{\gamma}(\boldsymbol{A}\hat{\boldsymbol{y}}^k - \boldsymbol{b}), \tag{3.6}$$

where $\hat{\boldsymbol{\theta}}^k = \boldsymbol{\theta}^k + \frac{k-1}{k+2}(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1})$, $\hat{\boldsymbol{y}}^k = \boldsymbol{y}^k + \frac{k-1}{k+2}(\boldsymbol{y}^k - \boldsymbol{y}^{k-1})$, and $\gamma = ||\boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T||$.

Using the first order optimality condition $\nabla_{\boldsymbol{y}}\bar{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\theta}) = \boldsymbol{0}$, the corresponding $\boldsymbol{y}^{k+1}$ is:

$$\boldsymbol{y}^{k+1} = -\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{\theta}^{k+1}. \tag{3.7}$$

Denote $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, ..., \boldsymbol{\theta}_N^T)^T$, and for $j = 1, ..., N$, $\boldsymbol{\theta}_j = (\theta_1^T(j), ..., \theta_m^T(j), \theta_g^T(j))^T$. Exploiting the step and subsystem separable structure of $\boldsymbol{A}$, $\boldsymbol{b}$, and the block diagonal structure of $\boldsymbol{R}$, (3.6) and (3.7) can be rewritten in a distributed manner as: for $\forall l = 1, ..., m$, and $j = 1, ..., N$,

$$\theta_l^{k+1}(j) = \hat{\theta}_l^k(j) + \frac{1}{\gamma}(\sum_{i=1}^{m} B_{li}u_i(j-1) + \sum_{i=1}^{m} A_{li}x_i(j-1) - x_l(j)), \tag{3.8a}$$

$$\theta_g^{k+1}(j) = \hat{\theta}_g^k(j) + \frac{1}{\gamma}(\sum_{i=1}^{m} A_j^i x_i(j) + \sum_{i=1}^{m} B_j^i u_i(j-1) - a_j), \tag{3.8b}$$

$$u_l^{k+1}(j-1) = -(R_{lj}^u)^{-1}((B_j^l)^T\theta_g^{k+1}(j) + \sum_{i=1}^{m} B_{il}^T\theta_l^{k+1}(j)), \tag{3.8c}$$

$$x_l^{k+1}(j) = -(R_{lj}^x)^{-1}((A_j^l)^T\theta_g^{k+1}(j) - \theta_l^{k+1}(j) + \sum_{i=1}^{m} A_{il}^T\theta_l^{k+1}(j+1)),\ j = 1, ..., N-1, \tag{3.8d}$$

$$x_l^{k+1}(j) = -(R_{lj}^x)^{-1}((A_j^l)^T\theta_g^{k+1}(j) - \theta_l^{k+1}(j)),\ j = N. \tag{3.8e}$$

Observe the summation appeared in (3.8a) - (3.8c), the compute efficiency of distributed manner can be further improved by introducing the following sets to skip the $\boldsymbol{0}$ item in summation.

$$\mathcal{N}_{xl} = \{i \in \{1, \ldots, m\} \mid A_{li} \neq \boldsymbol{0}\}, \tag{3.9a}$$

$$\mathcal{N}_{ul} = \{i \in \{1, \ldots, m\} \mid B_{li} \neq \boldsymbol{0}\}, \tag{3.9b}$$

$$\mathcal{M}_{xl} = \{i \in \{1, \ldots, m\} \mid A_{il} \neq \boldsymbol{0}\}, \tag{3.9c}$$

$$\mathcal{M}_{ul} = \{i \in \{1, \dots, m\} \mid B_{il} \neq \mathbf{0}\}, \tag{3.9d}$$

$$\mathcal{P}_j = \{i \in \{1, \dots, m\} \mid A^i_j \neq \mathbf{0}\}, \tag{3.9e}$$

$$\mathcal{Q}_j = \{i \in \{1, \dots, m\} \mid B^i_j \neq \mathbf{0}\}. \tag{3.9f}$$

As such, (3.8a) - (3.8d) can be written as: for $\forall l = 1, ..., m$, and $j = 1, ..., N$,

$$\theta^{k+1}_l(j) = \hat{\theta}^k_l(j) + \frac{1}{\gamma}(\sum_{i \in \mathcal{N}_{ul}} B_{li}u_i(j-1) + \sum_{i \in \mathcal{N}_{ul}} A_{li}x_i(j-1) - x_l(j)), \tag{3.10a}$$

$$\theta^{k+1}_g(j) = \hat{\theta}^k_g(j) + \frac{1}{\gamma}(\sum_{i \in \mathcal{P}_j} A^i_j x_i(j) + \sum_{i \in \mathcal{Q}_j} B^i_j u_i(j-1) - a_j), \tag{3.10b}$$

$$u^{k+1}_l(j-1) = -(R^u_{lj})^{-1}((B^l_j)^T\theta^{k+1}_g(j) + \sum_{i \in \mathcal{M}_{ul}} B^T_{il}\theta^{k+1}_l(j)). \tag{3.10c}$$

$$x^{k+1}_l(j) = -(R^x_{lj})^{-1}((A^l_j)^T\theta^{k+1}_g(j) - \theta^{k+1}_l(j) + \sum_{i \in \mathcal{M}_{xl}} A^T_{il}\theta^{k+1}_l(j+1)),\ j = 1, ..., N-1. \tag{3.10d}$$

Note that the summations exist in (3.10) can be further partitioned to row and column wise, which merely needs to apply the similar set definition as in (3.9), but will make the notations too tedious to follow by the reader. For this reason, the further partition of the matrix is spared, which, of course, is easily reachable in implementation.

## 3.3 The gradient based stopping condition and projection

In this section, the stopping condition to guarantee a predefined suboptimality $\epsilon$ in solving problem (3.4) is first studied. Then a linear projection that produces a primal $\epsilon$-suboptimal solution is proposed.

### 3.3.1 Stopping condition of $\epsilon$ suboptimal solution

As formulated in (3.5), the Lagrangian is a continuous quadratic function with positive definite hessian, thus differentiable. Given $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$, using the first order necessary optimality condition $\nabla_{\boldsymbol{y}}\bar{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\theta}) = \mathbf{0}$ gives:

$$\boldsymbol{y}_{\boldsymbol{\theta}} = -\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{\theta}, \tag{3.11}$$

Substituting (3.11) into (3.5) yields:

$$\bar{g}(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{\theta} - \boldsymbol{\theta}^T\boldsymbol{b}. \tag{3.12}$$

Based on the explicit expression of $\bar{g}(\boldsymbol{\theta})$ in (3.12), its first and second order gradient are respectively as:

$$\nabla\bar{g}(\boldsymbol{\theta}) = \boldsymbol{A}\boldsymbol{y}_\theta - \boldsymbol{b}, \tag{3.13}$$

$$\nabla^2\bar{g}(\boldsymbol{\theta}) = -\boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T. \tag{3.14}$$

**Lemma 1.** *Define* $\beta = \min \mathrm{eig}(\boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T)$*, in implementing iterative process* (3.6) (3.7)*, if*

$$||\boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b}||^2 \leq 2\beta\epsilon \tag{3.15}$$

*is satisfied, then* $(\boldsymbol{y}^k, \boldsymbol{\theta}^k)$ *is an* $\epsilon$ *dual solution of problem* (3.4)*.*

*Proof.* First, a convex optimization problem is introduced as:

$$F^* = \min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}), \tag{3.16}$$

where $F(\boldsymbol{\theta}) = -g(\boldsymbol{\theta})$ and $F^* = -\bar{g}^*$.

Subsequently, it gives:

$$||\nabla\bar{g}(\boldsymbol{\theta})||^2 = ||\nabla F(\boldsymbol{\theta})||^2 = ||\boldsymbol{A}\boldsymbol{y}_\theta - \boldsymbol{b}||^2, \tag{3.17}$$

$$||F(\boldsymbol{\theta}) - F^*|| = ||\bar{g}^* - \bar{g}(\boldsymbol{\theta})||. \tag{3.18}$$

Viewing (3.7) and (3.11), combined with (3.13), at each iteration it holds that:

$$\nabla\bar{g}(\boldsymbol{\theta}^k) = \boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b}, \tag{3.19}$$

Recall 9.1.2 of [8], for problem (3.16) to attain $\epsilon$-suboptimality such that $F(\boldsymbol{\theta}) - F^* \leq \epsilon$, the sufficient condition is:

$$||\nabla F(\boldsymbol{\theta})|| \leq (2\beta\epsilon)^{1/2}. \tag{3.20}$$

Using (3.17) (3.18) and (3.20), the proof can be concluded. □

### 3.3.2 From dual $\epsilon$ suboptimal solution to primal $\epsilon$ suboptimal solution

If $(\boldsymbol{y}_\theta, \boldsymbol{\theta})$ is an $\epsilon$ suboptimal solution of problem (3.4), it is not necessarily that $\boldsymbol{y} \in Y = \{\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}\}$, then an specific projection from $\boldsymbol{y}$ onto feasible set of problem (3.3) is needed. First, a matrix $\boldsymbol{F} \in \mathbb{R}^{ny \times (ny - n_e)}$ is introduced to satisfy $\boldsymbol{A}\boldsymbol{F} = \boldsymbol{0}$, and denote

$$\beta = \min \operatorname{eig}(\boldsymbol{F}^T \boldsymbol{R} \boldsymbol{F}). \tag{3.21}$$

Particularly, (3.21) could be fulfilled by the following treatment: given any $\boldsymbol{F'} \in \mathbb{R}^{ny \times (ny - n_e)}$ with $\boldsymbol{A}\boldsymbol{F'} = \boldsymbol{0}$ and $\min \operatorname{eig}(\boldsymbol{F'}^T \boldsymbol{R} \boldsymbol{F'}) \neq \beta$, let $b = \dfrac{\beta}{\min \operatorname{eig}(\boldsymbol{F'}^T \boldsymbol{R} \boldsymbol{F'})}$, then $\boldsymbol{F} = \sqrt{b}\boldsymbol{F'}$. Because the following holds:

$$\min \operatorname{eig}(\boldsymbol{F}^T \boldsymbol{R} \boldsymbol{F}) = \min \operatorname{eig}(b\boldsymbol{F'}^T \boldsymbol{R} \boldsymbol{F'}) = b(\min \operatorname{eig}(\boldsymbol{F'}^T \boldsymbol{R} \boldsymbol{F'})) = \beta.$$

Then, the feasible set $Y$ can be formulated as:

$$Y = \{\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}\} = \{\hat{\boldsymbol{y}} + \boldsymbol{F}\boldsymbol{t} \mid \boldsymbol{t} \in \mathbb{R}^{ny - n_e}\}, \tag{3.22}$$

this characterization is based on any $\hat{\boldsymbol{y}} \in Y$.

Next, inspired by linear projection operator $P_1$ in [12], $P_e$ from $\boldsymbol{y} \in \mathbb{R}^{ny}$ onto $Y$ is proposed as $P_e(\boldsymbol{y})$: $\boldsymbol{y} \mapsto \boldsymbol{y}_e$.

$$\boldsymbol{y}_e = \boldsymbol{y} - \boldsymbol{A}_p^T (\boldsymbol{A}_p \boldsymbol{A}_p^T)^{-1} (\boldsymbol{A}_p \boldsymbol{y} - \boldsymbol{b}_p), \tag{3.23}$$

where $\boldsymbol{A}_p = \boldsymbol{F}^T \boldsymbol{R} \oplus \boldsymbol{A}$, $\boldsymbol{b}_p = \boldsymbol{h} \oplus \boldsymbol{b}$, let $p = ny - n_e$, $\boldsymbol{h} \in \mathbb{R}^{ny - n_e}$ is any vector satisfying

$$||\boldsymbol{h}|| \leq ||\nabla \bar{g}(\boldsymbol{\theta})||. \tag{3.24}$$

**Lemma 2.** *If $(\boldsymbol{y}_\theta, \boldsymbol{\theta})$ is an $\epsilon$ suboptimal dual solution of problem (3.4), then $\boldsymbol{y}_e = P_e(\boldsymbol{y})$ is an $\epsilon$ primal solution of problem (3.3).*

*Proof.* First, it will be proved that $\boldsymbol{y}_e$ is a feasible solution of problem (3.3). Left multiplying the right side of (3.23) by $\boldsymbol{A}_p$, gives $\boldsymbol{A}_p \boldsymbol{y}_e = \boldsymbol{b}_p$, which could be partitioned as:

$$\boldsymbol{A}\boldsymbol{y}_e = \boldsymbol{b}, \tag{3.25}$$

$$\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{y}_e = \boldsymbol{h}. \tag{3.26}$$

By (3.25), it can be concluded that $\boldsymbol{y}_e \in Y_e$ is a primal feasible solution of problem (3.3).

Next, the $\epsilon$ suboptimality of $\boldsymbol{y}_e$ will be proved. By (3.22), Problem (3.3) is equivalent to:

$$\boldsymbol{J}^* = \min_{\boldsymbol{t}} \boldsymbol{J}(\boldsymbol{t}), \tag{3.27}$$

where $\boldsymbol{J}(\boldsymbol{t}) = \frac{1}{2}||\hat{\boldsymbol{y}} + \boldsymbol{F}\boldsymbol{t}||^2_{\boldsymbol{R}}$, and $\boldsymbol{J}^* = \bar{\mathcal{J}}^*$.

Accordingly,

$$\nabla \boldsymbol{J}(\boldsymbol{t}) = \boldsymbol{F}^T\boldsymbol{R}(\hat{\boldsymbol{y}} + \boldsymbol{F}\boldsymbol{t}),$$

$$\nabla^2 \boldsymbol{J}(\boldsymbol{t}) = \boldsymbol{F}^T\boldsymbol{R}\boldsymbol{F}.$$

By making $\boldsymbol{y}_e = \hat{\boldsymbol{y}} + \boldsymbol{F}\boldsymbol{t}_e$, it gives $\nabla \boldsymbol{J}(\boldsymbol{t}_e) = \boldsymbol{F}^T\boldsymbol{R}\boldsymbol{y}_e$.

Next, by (3.24):

$$||\nabla \boldsymbol{J}(\boldsymbol{t_e})||^2 = ||\boldsymbol{h}||^2 \leq ||\nabla \bar{g}(\boldsymbol{\theta})||^2 = ||\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}||^2. \tag{3.28}$$

As $\boldsymbol{J}(\boldsymbol{t}_e) = \mathcal{J}(\boldsymbol{y}_e)$, then by (3.15) and (3.21), it gives $||\mathcal{J}(\boldsymbol{y}_e) - \bar{\mathcal{J}}^*|| \leq \epsilon$. And this completes the proof. □

Integrating $\epsilon$ suboptimality stopping condition (3.15) and projection operator $P_e$, Algorithm 4 presents the procedures to generate $\epsilon$ primal solution for problem (3.3).

**Algorithm 4** Full Prediction Horizon Stopping Condition With Projection (FPH-P)

1: Initialize: $\boldsymbol{\theta}^0 = \boldsymbol{\theta}^{-1}$, $\epsilon$, $\beta$ and $k = 0$. $\boldsymbol{y}^{-1}$ and $\boldsymbol{y}^0$ are given by (3.7).
2: **while** (3.15) is not satisfied **do**
3: Update primal and dual variables by (3.8e) and (3.10)
4: $k \leftarrow k + 1$
5: **end while**
6: $\boldsymbol{y}_e^k = P_e(\boldsymbol{y}^k)$

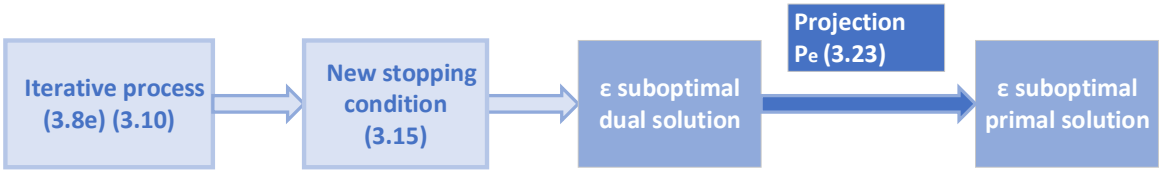


Figure 3.1 – Schematic diagram of Alg. 4 (FPH-P)

## 3.4 The first step focused stopping condition and projection

In the context of MPC, where only inputs of the first step in the current prediction horizon are applied, two questions in turn arise. Can (3.15) be transformed to stopping condition focused only on the first step? Is there a way to generate a feasible solution only using the first step elements while guarantee the $\epsilon$ suboptimality? These two questions are consecutively addressed in this section.

### 3.4.1 The first step focused stopping condition

Note that in the MPC context, the step separated structure existing in problem (3.3) for both objective function and constraints enables a step-based partition, which will serve as the basis of this section.

Here, the stopping condition (3.15) is converted into the first step oriented one, which consists of 2 steps. The first step is to partition the prediction horizon into two parts: the first step and interval from step 2 to $N$.

As a prerequisite, $\boldsymbol{A}$ and $\boldsymbol{b}$ are decomposed into block form as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{(1:1)} & \boldsymbol{0} \\ \boldsymbol{B}_{(2:N)} & \boldsymbol{A}_{(2:N)} \end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_{(1:1)} \\ \boldsymbol{b}_{(2:N)} \end{bmatrix}. \tag{3.29}$$

Based on (3.29), the gradient $\nabla \bar{g}(\boldsymbol{\theta})$ and iteration of $\boldsymbol{y}^k$ are partitioned as:

$$\boldsymbol{y}^k_{(1:1)} = -\boldsymbol{R}^{-1}_{(1:1)}(\boldsymbol{A}^T_{(1:1)}\boldsymbol{\theta}^k_{(1:1)} + \boldsymbol{B}^T_{(2:N)}\boldsymbol{\theta}^k_{(2:N)}), \tag{3.30}$$

$$\boldsymbol{y}^k_{(2:N)} = -\boldsymbol{R}^{-1}_{(2:N)}\boldsymbol{A}^T_{(2:N)}\boldsymbol{\theta}^k_{(2:N)}, \tag{3.31}$$

$$\nabla \bar{g}^k_{(1:1)} = \boldsymbol{A}_{(1:1)}\boldsymbol{y}^k_{(1:1)} - \boldsymbol{b}_{(1:1)}, \tag{3.32}$$

$$\nabla \bar{g}^k_{(2:N)} = \boldsymbol{B}_{(2:N)} \boldsymbol{y}^k_{(1:1)} + \boldsymbol{A}_{(2:N)} \boldsymbol{y}^k_{(2:N)} - \boldsymbol{b}_{(2:N)}, \tag{3.33}$$

where the variables in step partitioned gradient expression are omitted hereafter to lighten the notation.

The second step is to reconstruct (3.15): setting the required gradient norm from step 2 to N as 0. By doing so, a less demanding stopping condition for the first step with $\epsilon$ suboptimality guarantee is created, which enables a possible earlier termination of iterative process (3.6)-(3.7). As such, the early stopping condition is proposed as:

$$||\nabla \bar{g}^k_{(1:1)}||^2 \leq 2\beta\epsilon. \tag{3.34}$$

**Theorem 2.** *Let* $\bar{\boldsymbol{y}}_{(2:N)}$ *and* $\bar{\boldsymbol{\theta}}$ *be solved by the following equations:*

$$\boldsymbol{B}_{(2:N)} \boldsymbol{y}^k_{(1:1)} + \boldsymbol{A}_{(2:N)} \bar{\boldsymbol{y}}_{(2:N)} - \boldsymbol{b}_{(2:N)} = \boldsymbol{0}, \tag{3.35}$$

$$\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)} = -\boldsymbol{R}^{-1} \boldsymbol{A}^T \bar{\boldsymbol{\theta}}. \tag{3.36}$$

*Then, if* (3.34) *is satisfied in implementing iterative process* (3.6) (3.7)*,* $(\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)}, \bar{\boldsymbol{\theta}})$ *is an* $\epsilon$ *dual solution of problem* (3.4)*.*

*Proof.* By Assumption 3, all constraints of (3.3b) are linear independent, thus $\boldsymbol{A}$ is full row rank. Next, by partition (3.29), $\boldsymbol{A}_{(2:N)}$ is correspondingly full row rank. As such, by (3.35) (3.36), $\bar{\boldsymbol{y}}_{(2:N)}$ and $\bar{\boldsymbol{\theta}}$ could be solved explicitly. Since $\bar{\boldsymbol{y}}_{(2:N)}$ and $\bar{\boldsymbol{\theta}}$ are solvable, by (3.11) and (3.36), it gives

$$\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)} = \arg\min_{\boldsymbol{y}} \bar{\mathcal{L}}(\boldsymbol{y}, \bar{\boldsymbol{\theta}}). \tag{3.37}$$

which means $(\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)}, \bar{\boldsymbol{\theta}})$ is a dual solution of problem (3.4).

It remains to prove the following inequality.

$$\bar{g}^* - \bar{\mathcal{L}}(\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)}, \bar{\boldsymbol{\theta}}) \leq \epsilon.$$

By Lemma 1, it needs to have:

$$||\nabla \bar{g}(\bar{\boldsymbol{\theta}})||^2 \leq 2\beta\epsilon.$$

By (3.32) (3.33) and (3.36),

$$||\nabla \bar{g}(\bar{\boldsymbol{\theta}})||^2 = ||\nabla \bar{g}^k_{(1:1)}||^2 + ||\boldsymbol{B}_{(2:N)}\boldsymbol{y}^k_{(1:1)} + \boldsymbol{A}_{(2:N)}\bar{\boldsymbol{y}}_{(2:N)} - \boldsymbol{b}_{(2:N)}||^2.$$

The proof can be concluded by (3.35) and (3.34). □

**Lemma 3.** *In implementing iterative process* (3.6)-(3.7)*, let* $k_1$ *and* $k_2$ *be defined as:*

$$k_1 = \inf\{k \mid ||\nabla \bar{g}^k_{(1:1)}||^2 \leq 2\beta\epsilon\}, \tag{3.38}$$

$$k_2 = \inf\{k \mid ||\nabla \bar{g}^k||^2 \leq 2\beta\epsilon\}, \tag{3.39}$$

*then* $k_1 \leq k_2$.

*Proof.* Since for $\forall k$, $||\nabla \bar{g}^k||^2 = ||\nabla \bar{g}^k_{(1:1)}||^2 + ||\nabla \bar{g}^k_{(2:N)}||^2$, and $||\nabla \bar{g}^k_{(2:N)}||^2 \geq 0$, then it gives

$$||\nabla \bar{g}^{k_2}_{(1:1)}||^2 \leq ||\nabla \bar{g}^{k_2}||^2 \leq 2\beta\epsilon.$$

As a consequence, by (3.38), $k_1 \leq k_2$ holds. And this completes the proof. □

### 3.4.2 The first step focused projection

In this subsection, the projection operator $P_{e,(1:1)}$ from $\boldsymbol{y}_{(1:1)} \in \mathbb{R}^{ny_{(1:1)}}$ onto $Y_{(1:1)}$ is specified as $P_{e,(1:1)}(\boldsymbol{y}_{(1:1)})$: $\boldsymbol{y}_{(1:1)} \mapsto \boldsymbol{y}_{e,(1:1)}$.

$$\boldsymbol{y}_{e,(1:1)} = \boldsymbol{y}_{(1:1)} - \boldsymbol{A}^T_{p,(1:1)}(\boldsymbol{A}_{p,(1:1)}\boldsymbol{A}^T_{p,(1:1)})^{-1}(\boldsymbol{A}_{p,(1:1)}\boldsymbol{y}_{(1:1)} - \boldsymbol{b}_{p,(1:1)}), \tag{3.40}$$

where $\boldsymbol{A}_{p,(1:1)} = \boldsymbol{F}^T_{(1:1)}\boldsymbol{R}_{(1:1)} \oplus \boldsymbol{A}_{(1:1)}$, $\boldsymbol{b}_{p,(1:1)} = \boldsymbol{h}_{(1:1)} \oplus \boldsymbol{b}_{(1:1)}$, $\nabla \bar{g}_{(1:1)}(\boldsymbol{\theta}_{(1:1)}) = \boldsymbol{A}_{(1:1)}\boldsymbol{y}_{(1:1)} - \boldsymbol{b}_{(1:1)}$, let $p_{(1:1)} = ny_{(1:1)} - n_{e,(1:1)}$ and $\boldsymbol{h}_{(1:1)} \in \mathbb{R}^{p_{(1:1)}}$ is any vector satisfying $||\boldsymbol{h}_{(1:1)}|| \leq ||\nabla \bar{g}_{(1:1)}(\boldsymbol{\theta}_{(1:1)})||$.

**Lemma 4.** *If* $\boldsymbol{y}^k_{(1:1)}$ *satisfies* (3.34)*, then* $\boldsymbol{y}^k_{e,(1:1)} = P_{e,(1:1)}(\boldsymbol{y}^k_{(1:1)})$ *is the first step components of an* $\epsilon$ *primal solution of problem* (3.3)*.*

*Proof.* By Theorem 2, $(\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)}, \bar{\boldsymbol{\theta}})$ is an $\epsilon$ dual solution of problem (3.4). Implementing projection $P_e$, by Lemma 2, an $\epsilon$ primal solution of problem (3.3) is $\bar{\boldsymbol{y}}_e = P_e(\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)})$, which could be partitioned as $\bar{\boldsymbol{y}}_e = \bar{\boldsymbol{y}}_{e,(1:1)} \oplus \bar{\boldsymbol{y}}_{e,(2:N)}$.

By (3.25) (3.29), for feasibility fulfillment:

$$\boldsymbol{A}_{(1:1)}\bar{\boldsymbol{y}}_{e,(1:1)} = \boldsymbol{b}_{(1:1)}, \tag{3.41}$$

$$\boldsymbol{A}_{(2:N)}\bar{\boldsymbol{y}}_{e,(2:N)} + \boldsymbol{B}_{(2:N)}\bar{\boldsymbol{y}}_{e,(1:1)} = \boldsymbol{b}_{(2:N)}. \tag{3.42}$$

By (3.28) (3.29), for $\epsilon$ suboptimality fulfillment:

$$||\boldsymbol{F}_{(1:1)}^T\boldsymbol{R}_{(1:1)}\bar{\boldsymbol{y}}_{e,(1:1)}||^2 \leq ||g_{(1:1)}^k||^2 = ||\boldsymbol{A}_{(1:1)}\boldsymbol{y}_{(1:1)}^k - \boldsymbol{b}_{(1:1)}||^2, \tag{3.43}$$

$$||\boldsymbol{F}_{(2:N)}^T\boldsymbol{R}_{(2:N)}\bar{\boldsymbol{y}}_{e,(2:N)}||^2 = 0. \tag{3.44}$$

Let $\boldsymbol{y}_{e,(1:1)}^k = P_{e,(1:1)}(\boldsymbol{y}_{(1:1)}^k)$, $\boldsymbol{y}_{e,(1:1)}^k$ satisfies (3.41) and (3.43). Then substituting $\bar{\boldsymbol{y}}_{e,(1:1)}$ by $\boldsymbol{y}_{e,(1:1)}^k$ in (3.42), a linear equation group (with $\boldsymbol{y}_{e,(2:N)}^k$ being variable to be solved) can be formed as:

$$\begin{cases} \boldsymbol{A}_{(2:N)}\boldsymbol{y}_{e,(2:N)}^k + \boldsymbol{B}_{(2:N)}\boldsymbol{y}_{e,(1:1)}^k = \boldsymbol{b}_{(2:N)}, \\ \boldsymbol{F}_{(2:N)}^T\boldsymbol{R}_{(2:N)}\boldsymbol{y}_{e,(2:N)}^k = \boldsymbol{0}. \end{cases} \tag{3.45}$$

As $\boldsymbol{y}_{e,(2:N)}^k \in \mathbb{R}^{ny_{(2:N)}}$, $\boldsymbol{A}_{(2:N)} \in \mathbb{R}^{n_{e,(2:N)} \times ny_{(2:N)}}$, $\boldsymbol{F}_{(2:N)} \in \mathbb{R}^{ny_{(2:N)} \times p_{(2:N)}}$ and $\boldsymbol{R}_{(2:N)} \in \mathbb{S}_{++}^{ny_{(2:N)}}$, by Assumption 3, $\text{rank}(\boldsymbol{A}_{(2:N)}) = n_{e,(2:N)}$, $\text{rank}(\boldsymbol{F}_{(2:N)}^T\boldsymbol{R}_{(2:N)}) = p_{(2:N)}$ , thus (3.45) has the unique solution of $\boldsymbol{y}_{e,(2:N)}^k$.

Consequently, $\boldsymbol{y}_{e,(1:1)}^k \oplus \boldsymbol{y}_{e,(2:N)}^k$ is indeed an $\epsilon$ primal solution of problem (3.3) by Definition 1, of which $\boldsymbol{y}_{e,(1:1)}^k$ is the first step components. And this completes the proof. □

Algorithm 5 depicts the mechanism combining the first step focused stopping condition and the first step projection. Note that in this section, the presence of $\bar{\boldsymbol{y}}_{(2:N)}$, $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{y}_{e,(2:N)}^k$

**Algorithm 5** First Step Stopping Condition With Projection (FS-P)

1: Initialize: $\boldsymbol{\theta}^0 = \boldsymbol{\theta}^{-1}$, $\epsilon$, $\beta$ and $k = 0$. $\boldsymbol{y}^{-1}$ and $\boldsymbol{y}^0$ are given by (3.7).
2: **while** (3.34) is not satisfied **do**
3: Update primal and dual variables by (3.8e) and (3.10)
4: $k \leftarrow k + 1$
5: **end while**
6: $\boldsymbol{y}_{e,(1:1)}^k = P_{e,(1:1)}(\boldsymbol{y}_{(1:1)}^k)$

are purely for establishing mathematical proof, only $\boldsymbol{y}_{(1:1)}^k$ and $\boldsymbol{y}_{e,(1:1)}$ need to be computed in implementing FS-P.

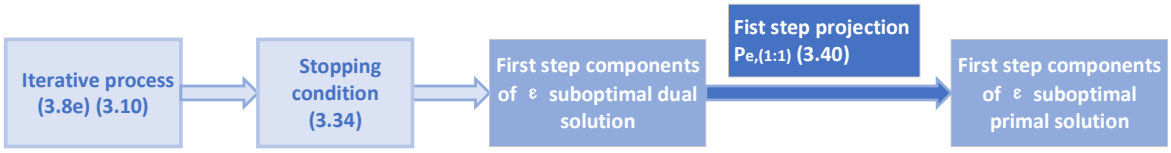


Figure 3.2 – Schematic diagram of Alg. 5 (FS-P)

Table 3.1 – Suboptimality performance comparison among FS, FS-P, FPH and FPH-P. The magnitude of $1 \times 10^{-3}$, $1 \times 10^{-4}$ and $1 \times 10^{-5}$ are omitted from the results according to the relative suboptimality referred for space-saving. As a result, any result showed with absolute value less than 1 means predefined suboptimality is guaranteed.

| Predefined Rel. $\epsilon$ | Alg. | Ave. Rel. Error | | | | | Max. Rel. Error | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Prediction Horizon | | | | | Prediction Horizon | | | | |
| | | 10 | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 |
| $10^{-3}$ | FPH | -0.58 | -0.48 | -0.47 | -0.44 | -0.46 | -0.94 | -0.91 | -0.92 | -0.89 | 0.95 |
| | FS | -0.48 | -0.56 | -0.58 | -0.67 | -0.70 | -0.91 | -0.95 | -0.96 | -0.99 | -0.98 |
| | FPH-P | 0.89 | 0.87 | 0.86 | 0.83 | 0.85 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| | FS-P | 0.83 | 0.85 | 0.84 | 0.86 | 0.85 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| $10^{-4}$ | FPH | -0.54 | -0.51 | -0.51 | -0.50 | -0.50 | -0.91 | -0.93 | -0.93 | -0.97 | -0.92 |
| | FS | -0.55 | -0.50 | -0.50 | -0.50 | -0.47 | -0.98 | -0.97 | -0.94 | -0.98 | -0.99 |
| | FPH-P | 0.88 | 0.87 | 0.88 | 0.86 | 0.85 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| | FS-P | 0.85 | 0.83 | 0.83 | 0.86 | 0.85 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| $10^{-5}$ | FPH | -0.57 | -0.56 | -0.54 | -0.51 | -0.50 | -0.92 | -0.94 | -0.96 | -0.94 | -0.92 |
| | FS | -0.53 | -0.48 | -0.45 | -0.46 | -0.50 | -0.92 | -0.96 | -0.94 | -0.98 | -0.91 |
| | FPH-P | 0.89 | 0.89 | 0.87 | 0.84 | 0.85 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| | FS-P | 0.81 | 0.81 | 0.80 | 0.81 | 0.81 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |

## 3.5 Numerical experiments

In this section, FPH-P and FS-P are tested under 5 prediction horizons, from 10 to 50 with the incremental interval of 10. Of each prediction horizon, 100 independent randomly generated numerical experiments are carried out using MATLAB 2018b on a Windows 10 PC with 2.20 GHz Core i7-8750H CPU and 16GB RAM.

The tested system, with $A_{li}$ and $B_{li}$ randomly generated by MATLAB command `drss`, consists of 5 subsystems, each of which contains 2 inputs and 2 states. More in details, each element of $\boldsymbol{x}_0$ is randomly drawn from uniform distribution $[-0.5, 0.5]$. The global equality constraints are $\sum_{i=1}^{m} u_l(j) = 1$, for $j = 0, ..., N-1$. The penalty matrix $\boldsymbol{R} = \boldsymbol{I}$, and the predefined relative suboptimality [1] tested are $1 \times 10^{-3}$, $1 \times 10^{-4}$ and $1 \times 10^{-5}$.

Particularly, for projection treatment, making the first element of $\boldsymbol{h}$ equal to $||\nabla \bar{g}(\boldsymbol{\theta}^k)||$ and rest elements equal 0 in Step 6 of FPH-P can generated $\boldsymbol{y}_e^k$; and make the first element of $\boldsymbol{h}_{(1:1)}$ equal to $||\nabla \bar{g}_{(1:1)}^k||$ and rest elements equal 0 in Step 6 of FS-P to get $\boldsymbol{y}_{e,(1:1)}^k$. Subsequently, $\boldsymbol{y}_{e,(2:N)}^k$ is solved by (3.45) using $\boldsymbol{y}_{e,(1:1)}^k$.

In Table 3.1, FPH refers to $g(\boldsymbol{\theta}^k)$ when (3.15) in FPH-P is satisfied. Specifically, FS refers to the full length prediction objective value $g(\bar{\boldsymbol{\theta}})$, and $\bar{\boldsymbol{\theta}}$ is solved by (3.35) (3.36) when (3.34) is satisfied. The benchmark value $\bar{\mathcal{J}}^*$ for relative error comparison is solved by commercial optimization solver MOSEK programmed in platform YALMIP[31].

Table 3.1 shows that the predefined suboptimality of all $N$ cases is guaranteed by implementing FS, FPH, and their projections. The discrepancies of both average and maximal relative error between FS and FPH are comparably minor, so as that between FS-P and FPH-P, which suggests that using the first step stopping condition does not impact the fulfillment of feasibility and $\epsilon$ suboptimality.

In Fig. 3.3, by projection mechanism, the trajectory of FS-P and FPH-P are closer to the optimal solution than FS and FPH. Furthermore, the later the step appeared in the input sequence of FS-P and FPH-P, the tighter the gap between them and the optimal solution.

From Fig. 3.6, Fig. 3.5 and Fig. 3.4, statistically, FS consumes significantly fewer iterations compared to FPH in majority tests of all $N$ cases for all relative suboptimality. Note that, ratio as 1 denotes that $k_1 = k_2$ in Lemma 3, which is the worst case that could happen to FS in terms of iteration number. Generally, the ratio of FS to FPH in computation time is even smaller than that in the iteration number. Observing (3.34)

1. The relative suboptimality is computed as suboptimality divided by $\bar{\mathcal{J}}^*$.

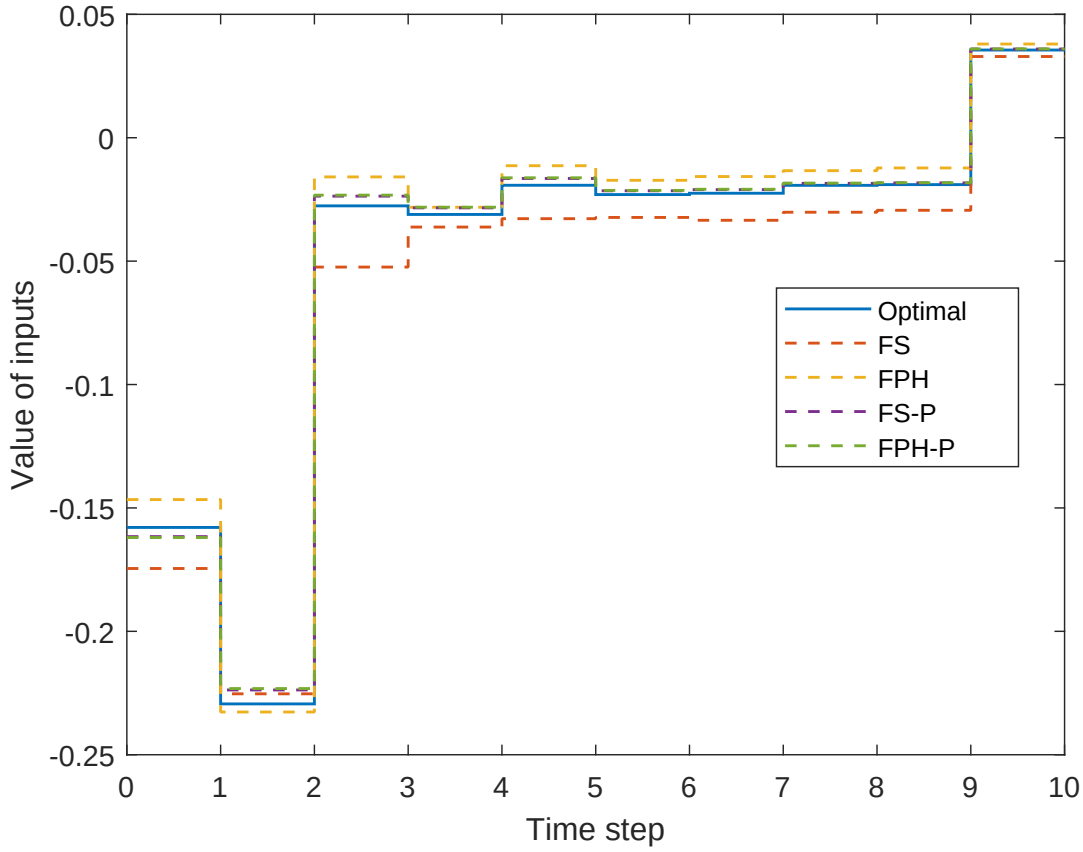


Figure 3.3 – Input sequence comparison of one subsystem in a test among optimal solution, FS, FPH, FS-P and FPH-P
$N = 10$ and predefined relative $\epsilon$ as $1 \times 10^{-3}$, $\boldsymbol{y}^k_{(1:1)} \oplus \bar{\boldsymbol{y}}_{(2:N)}$ and $\boldsymbol{y}^k_{e,(1:1)} \oplus \boldsymbol{y}^k_{e,(2:N)}$ are presented respectively for FS and FS-P, note that only inputs of the first time step would be applied.

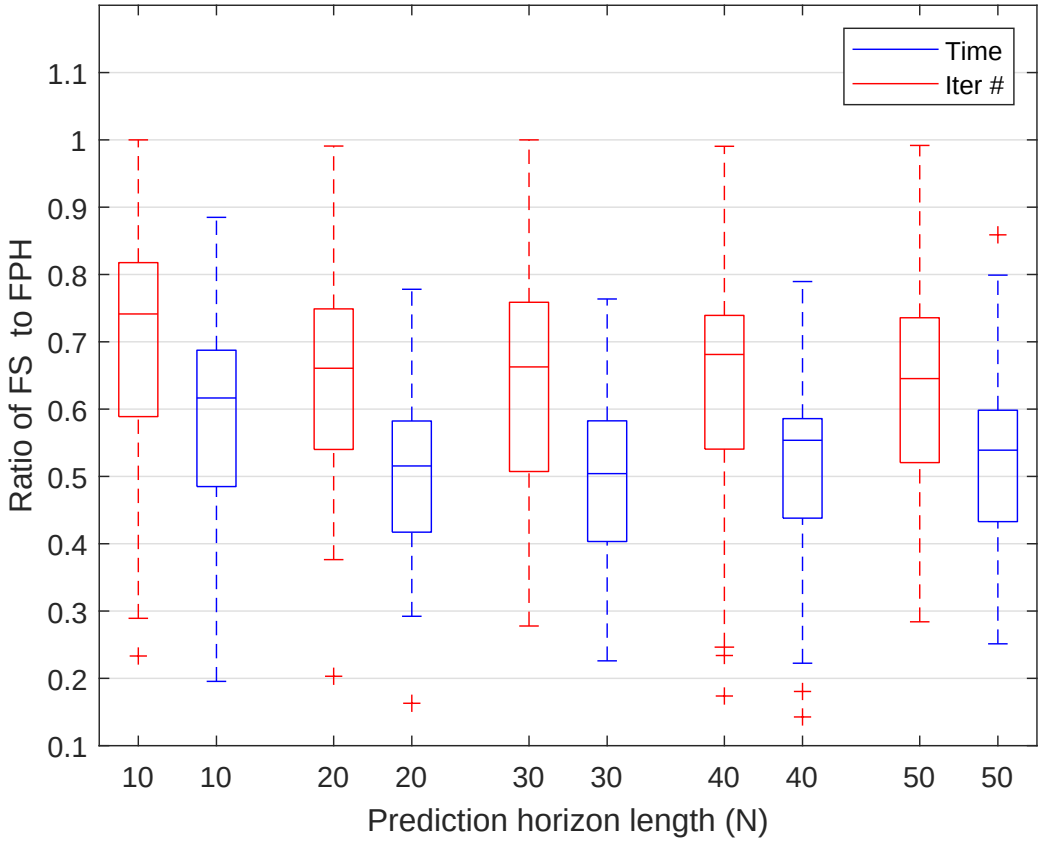


Figure 3.4 – Iteration number and computation time ratio of FS to FPH with predefined relative $\epsilon$ as $1 \times 10^{-5}$
Sample value exceeded $+/-2.7\sigma$ shows as whisker. Sample value less, greater than or equals to 1 means FS spends less, more or the same time/iterations as FPH in the same test. The lower value, the better performance of FS.

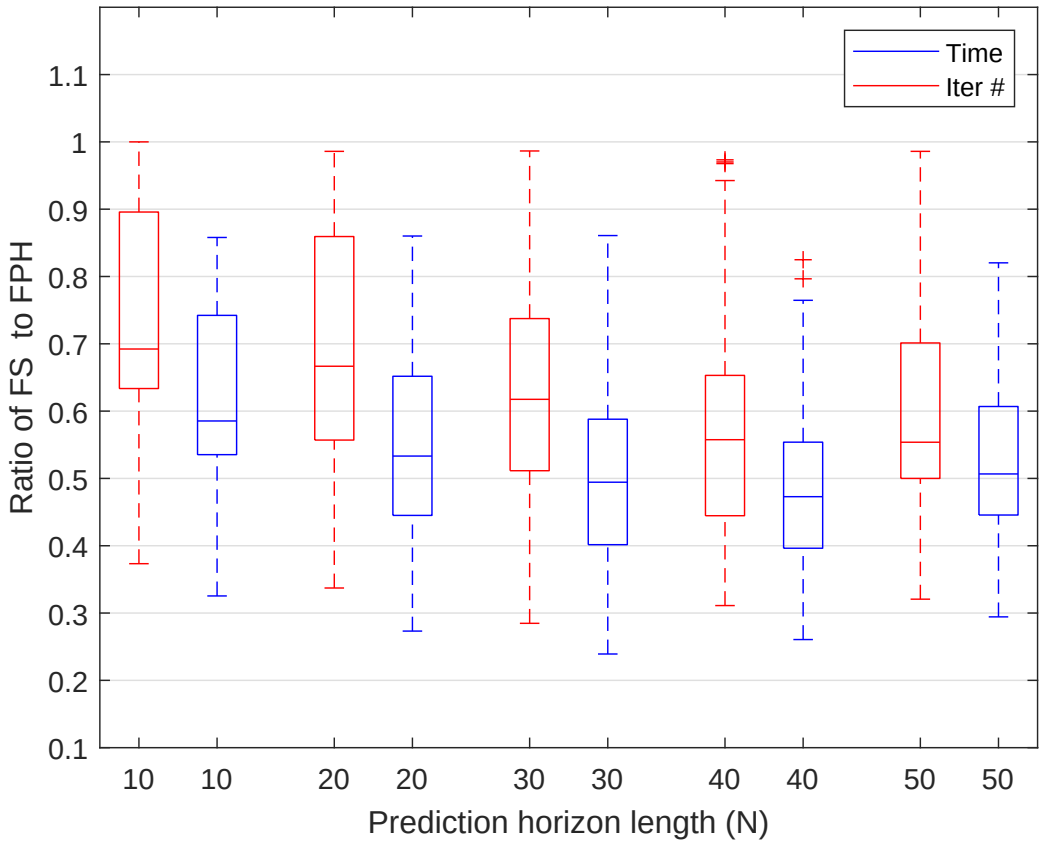


Figure 3.5 – Iteration number and computation time ratio of FS to FPH with predefined relative $\epsilon$ as $1 \times 10^{-4}$
Sample value exceeded $+/-2.7\sigma$ shows as whisker. Sample value less, greater than or equals to 1 means FS spends less, more or the same time/iterations as FPH in the same test. The lower value, the better performance of FS.

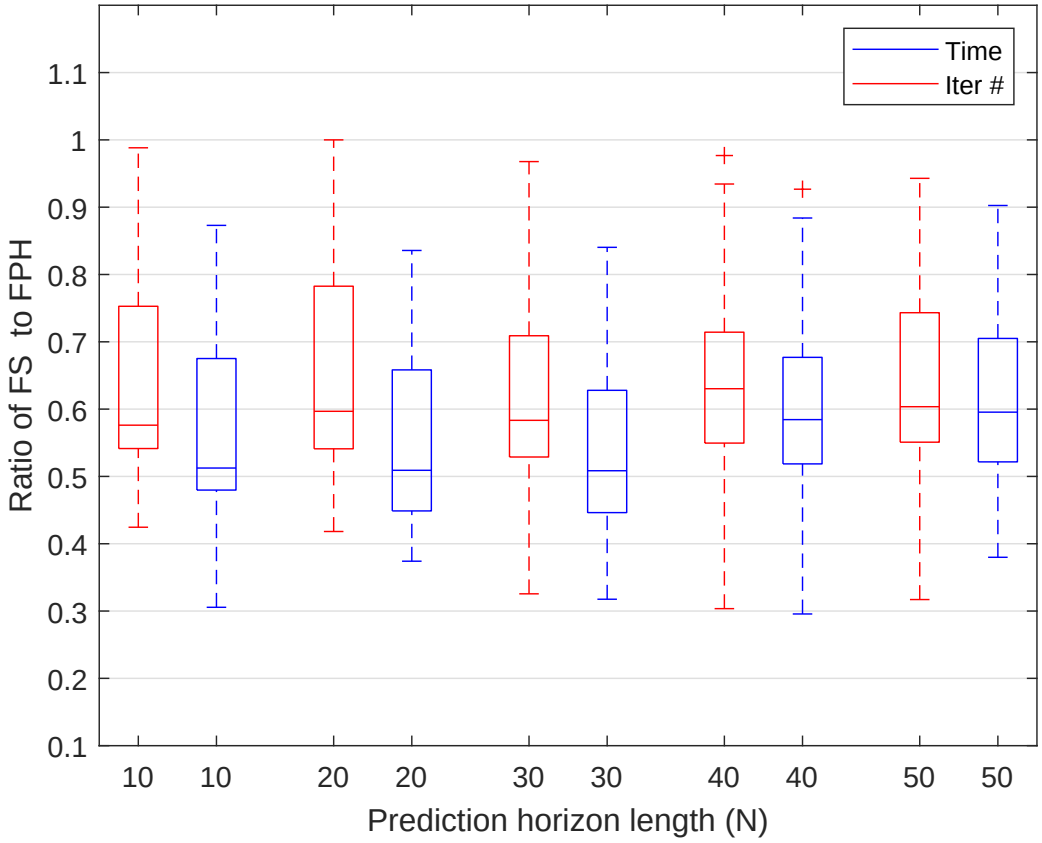


Figure 3.6 – Iteration number and computation time ratio of FS to FPH with predefined relative $\epsilon$ as $1 \times 10^{-3}$
Sample value exceeded $+/-2.7\sigma$ shows as whisker. Sample value less, greater than or equals to 1 means FS spends less, more or the same time/iterations as FPH in the same test. The lower value, the better performance of FS.

and (3.15), at each iteration, FS only need to calculate gradient for the first step. At the same time, the full prediction horizon calculation is required of FPH, resulting in an even larger time advantage of FS as $N$ increases.

## 3.6 Conclusion

In this chapter, in tackling optimization resulted from MPC, projection onto the primal feasible set with $\epsilon$-suboptimality guarantee has been proposed based on an $\epsilon$ dual solution. The early stopping condition in the MPC context has been demonstrated through a step-based partition technique by focusing on the first step components. The first step focused projection with $\epsilon$-suboptimality and feasibility guarantee has also been introduced to generate applicable inputs for the system.

Through random numerical experiments, the $\epsilon$-suboptimality condition has been verified for both algorithms. Due to less demanding stopping condition and computation burdens, the first step focused algorithm has generally outperformed the full prediction horizon algorithm largely in iteration number and computation time.

In §4, the general convex inequality constraints will be included in the problem setting, with which suboptimality criterion for both full sequence and the first step will be derived. In contrast to Chapter 1 and 2, the second order method will be studied for optimization resulted from MPC to achieve faster convergence.

Chapter 4

# $\epsilon$ SUBOPTIMALITY BASED ACCELERATED TERMINATION FOR MPC USING PRIMAL DUAL INTERIOR POINT METHOD

In this chapter, by employing the logarithmic barrier function, the inequality constraints can be converted into an item in the objective function of the MPC optimization problem, making it only an equality constrained problem. By introducing the modified KKT condition, the suboptimality criterion can be designed in the presence of bounded primal and dual infeasibility, which is a more flexible condition for the iterative process to quit. Targeting at an accelerated termination, the step-wise structure of MPC is exploited. The first step focused criterion is invented to guarantee the predefined suboptimality and the bounded primal and dual infeasibility.

## 4.1 Introduction

The iterative method adopted in this chapter, primal dual interior point method[35][67], with quadratic convergence rate[68] and good scalability, is deemed strongly favorable over the gradient and steepest descent methods[8].

Plentiful work has been made to apply primal dual interior point method in MPC formulation in searching for a suboptimal solution, e.g., the inequality constraints are converted into equality constraints by adding non-negative slack variables in search of Newton step [35], which is retrofitted to peculiar MPC KKT system structure by block elimination [50] [66]; a fast computation of Newton step based on Cholesky factorization is applied to linear inequality constrained system, achieving an order of magnitude decrease of complexity[63], which in [19] has been further decreased by a low rank matrix forward substitution scheme, and can be extended to quadratic inequality constraints.

Given characteristics of MPC that only first step inputs are applied to the system, by

exploiting the step separable structure of system dynamics and inequality constraints, the first step focused criterion is designed to ensure the predefined suboptimality with only the first step component being determined. More in details, the first step components alone can be tested by stopping conditions, as long as the remaining steps (from step 2 to N in the prediction horizon) can form the components of the optimal solution of full sequence. In this way, an accelerated termination of the iterative process compared to the traditional primal dual interior point method is possible.

The main contribution of this chapter is threefold. First, the first step focused criterion suited for the interior point method is proposed to generate a suboptimal solution for the MPC problem. Second, the superiority of the first step focused criterion in terms of iteration number is demonstrated both theoretically and experimentally. Third, the whole methodology in this chapter accommodates the general convex inequality constraints, unlike many methods that can only work with linear inequality constraints.

This chapter is organized as follows. §4.2 sets up the optimization problem and fundamentals of barrier function based approximation. §4.3 introduces the primal dual interior point method for general convex optimization resulted from MPC. §4.4 proposes the first step focused stopping criterion, whose effectiveness is demonstrated. Numerical experiments and results discussions are presented in §4.5. And conclusions are given in §4.6.

## 4.2 Problem statement and fundamentals

In this section, the control system and MPC optimization problem are presented. To eliminate the inequality constraints, the logarithmic barrier function is introduced to formulate the approximate problem to the original problem, of which the approximation error is also given.

### 4.2.1 Problem statement

The coupling dynamics and global equality constraints are formed as: for $\forall l = 1, ..., m$, and $j = 1, ..., N$,

$$x_l(j) = \sum_{i=1}^{m}(A_{li}x_i(j-1) + B_{li}u_i(j-1)), \tag{4.1}$$

$$\sum_{l=1}^{m} A_j^l x_l(j) + \sum_{l=1}^{m} B_j^l u_l(j-1) = a_j, \tag{4.2}$$

where $x_l(j) \in \mathbb{R}^{nx_l}$ and $u_l(j-1) \in \mathbb{R}^{nu_l}$ are states at step $j$ and inputs at step $j-1$ of $l$-th subsystem, $A_{li} \in \mathbb{R}^{nx_l \times nx_i}$, $B_{li} \in \mathbb{R}^{nx_l \times nu_i}$ are system matrix, and $x_l(0) = \bar{x}_l$, $\bar{x}_l \in \mathbb{R}^{nx_l}$ is the initial state of $l$-th subsystem, $A_j^l \in \mathbb{R}^{na_j \times nx_l}$, $B_j^l \in \mathbb{R}^{na_j \times nu_l}$ and $a_j \in \mathbb{R}^{na_j}$.

The inequality constraints considered in this chapter is:

$$f_{i,j}(\boldsymbol{x}_j, \boldsymbol{u}_{j-1}) \leq 0,\ j = 1, ..., N,\ i = 1, ..., n_j, \tag{4.3}$$

where $f_{i,j} : \mathbb{R}^{nx} \times \mathbb{R}^{nu} \to \mathbb{R}$ is assumed to be convex and twice continuous differentiable, $n_j \in \mathbb{N}_{>0}$ denotes the number of inequality constraints at step $j$, for the consistency of total inequality constraints number defined in (1.3), it holds that $n_1 + ... + n_N = n$, and for $j = 1, ..., N$, composition of $\boldsymbol{x}_j$ and $\boldsymbol{u}_{j-1}$ can be referred to (1.3).

Note that the total number of $n$ of inequality constraints is crucial to building suboptimality criterion, which will be shown in the following subsection.

The MPC resulted problem can be compactly formulated as below:

$$\tilde{\mathcal{J}}^* = \min_{\boldsymbol{y}}\ \mathcal{J}(\boldsymbol{y}), \tag{4.4a}$$

$$s.t.\ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \tag{4.4b}$$

$$\tilde{\boldsymbol{f}}(\boldsymbol{y}) \leq \boldsymbol{0}, \tag{4.4c}$$

where (4.4b) is taken from (1.5b), composition of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{y}$ can be referred to (3.1), (3.2) and (1.5) respectively, $\tilde{\boldsymbol{f}} : \mathbb{R}^{ny} \to \mathbb{R}^n$ with $\tilde{\boldsymbol{f}}(\boldsymbol{y}) = (\tilde{\boldsymbol{f}}_1(\boldsymbol{y}_1)^T, ..., \tilde{\boldsymbol{f}}_N(\boldsymbol{y}_N)^T)^T$, and $\forall\ i = 1, ..., n_j,\ j = 1, ..., N,\ \tilde{\boldsymbol{f}}_j(\boldsymbol{y}_j) = (\tilde{f}_{1,j}(\boldsymbol{y}_j)^T, ..., \tilde{f}_{n_j,j}(\boldsymbol{y}_j)^T)^T$, $\tilde{f}_{i,j}(\boldsymbol{y}_j) = f_{i,j}(\boldsymbol{x}_j, \boldsymbol{u}_{j-1})$, $\tilde{f}_{i,j} : \mathbb{R}^{nx+nu} \to \mathbb{R}$.

### 4.2.2 Barrier function based approximation

In this subsection, based on Chapter 11 of [8], the approximation problem of (4.4) is formulated using barrier function, whose approximation bound is derived subsequently.

First, the logarithmic barrier function $\boldsymbol{\phi}$ of $\tilde{\boldsymbol{f}}$ is introduced as:

$$\boldsymbol{\phi}(\boldsymbol{y}) = -\sum_{j=1}^{N}\sum_{i=1}^{n_j} \log(-\tilde{f}_{i,j}(\boldsymbol{y}_j)), \tag{4.5}$$

where the domain of $\boldsymbol{\phi}$ is $\{\boldsymbol{y} \in \mathbb{R}^{ny} \mid \tilde{\boldsymbol{f}}(\boldsymbol{y}) < \boldsymbol{0}\}$.

The gradient and hessian of $\boldsymbol{\phi}$ are as:

$$\nabla\boldsymbol{\phi}(\boldsymbol{y}) = -\sum_{j=1}^{N}\sum_{i=1}^{n_j}\frac{\nabla\tilde{f}_{i,j}(\boldsymbol{y}_j)}{\tilde{f}_{i,j}(\boldsymbol{y}_j)}, \tag{4.6}$$

$$\nabla^2\boldsymbol{\phi}(\boldsymbol{y}) = \sum_{j=1}^{N}\sum_{i=1}^{n_j}\frac{\nabla\tilde{f}_{i,j}(\boldsymbol{y}_j)\nabla\tilde{f}_{i,j}(\boldsymbol{y}_j)^T}{\tilde{f}_{i,j}(\boldsymbol{y}_j)^2} - \sum_{j=1}^{N}\sum_{i=1}^{n_j}\frac{\nabla^2\tilde{f}_{i,j}(\boldsymbol{y}_j)}{\tilde{f}_{i,j}(\boldsymbol{y}_j)}. \tag{4.7}$$

Now, the approximate problem of (4.4) is formulated as:

$$\min_{\boldsymbol{y}}\ t\mathcal{J}(\boldsymbol{y}) + \boldsymbol{\phi}(\boldsymbol{y}),\ s.t.\ (4.4\text{b}), \tag{4.8}$$

where $t \in \mathbb{R}_{>0}$ is a parameter of approximation (the quantitative analysis will be illustrated later), and the optimal solution is denoted as $\boldsymbol{y}^*(t)$.

Next, the Lagrangian associated with problem (4.8) is introduced as:

$$\bar{L}(\boldsymbol{y},\boldsymbol{\theta}) = t\mathcal{J}(\boldsymbol{y}) + \boldsymbol{\phi}(\boldsymbol{y}) + \boldsymbol{\theta}^T(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}), \tag{4.9}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$ is the dual variable associated with equality constraint (4.4b).

A point $(\boldsymbol{y}^*(t), \boldsymbol{\theta}^*(t))$ is said to be primal and dual optimal of problem (4.8), if it satisfies the KKT condition equations listed below, which are necessary and sufficient condition for optimality.

$$t\boldsymbol{R}\boldsymbol{y}^*(t) + \nabla\boldsymbol{\phi}(\boldsymbol{y}^*(t)) + \boldsymbol{A}^T\hat{\boldsymbol{\theta}}^*(t) = \boldsymbol{0}, \tag{4.10a}$$

$$\boldsymbol{A}\boldsymbol{y}^*(t) - \boldsymbol{b} = \boldsymbol{0}. \tag{4.10b}$$

Subsequently, a dual variable $\lambda^*_{i,j}(t) \in \mathbb{R}$ is defined by

$$\lambda^*_{i,j}(t) = -\frac{1}{t\tilde{f}_{i,j}(\boldsymbol{y}^*_j)},\ i = 1,...,n_j,\ j = 1,...,N, \tag{4.11}$$

where $\boldsymbol{y}^* = ((\boldsymbol{y}^*_1)^T,...,(\boldsymbol{y}^*_N)^T)^T$, and $\lambda^*_{i,j}(t)$ must be positive, since $\tilde{f}_{i,j}(\boldsymbol{y}^*_j) < 0$ must be fulfilled.

Dividing $t$ to both sides of (4.10a) by (4.6), it gives:

$$\boldsymbol{R}\boldsymbol{y}^*(t) + \nabla\tilde{\boldsymbol{f}}(\boldsymbol{y}^*(t))\boldsymbol{\lambda}^*(t) + \boldsymbol{A}^T\boldsymbol{\theta}^*(t) = \boldsymbol{0}, \tag{4.12}$$

where $\boldsymbol{\theta}^*(t) \in \mathbb{R}^{n_e}$ , $\boldsymbol{\theta}^*(t) = \hat{\boldsymbol{\theta}}^*(t)/t$, $\boldsymbol{\lambda}^*(t) = ((\boldsymbol{\lambda}^*_1(t))^T,...,(\boldsymbol{\lambda}^*_N(t))^T)^T$, $\boldsymbol{\lambda}^*(t) \in \mathbb{R}^n$, and for

$j = 1, ..., N$, $\boldsymbol{\lambda}_j^*(t) = ((\lambda_{1,j}^*(t))^T, ..., (\lambda_{n_j,j}^*(t))^T)^T$, and $\boldsymbol{\lambda}_j^*(t) \in \mathbb{R}^{n_j}$.

Alternatively, (4.12) can be interpreted as the minimization over $\boldsymbol{y}$ of the following Lagrangian (with solution as $(\boldsymbol{y}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\theta}^*(t))$):

$$\hat{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \mathcal{J}(\boldsymbol{y}) + \boldsymbol{\lambda}^T \tilde{\boldsymbol{f}}(\boldsymbol{y}) + \boldsymbol{\theta}^T(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}), \tag{4.13}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^n$ is the dual variable associated with inequality constraint (4.4c). Consider the dual function associated with problem (4.4):

$$\hat{g}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \min_{\boldsymbol{y}} \hat{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta}). \tag{4.14}$$

Then for $\forall t > 0$, $(\boldsymbol{\lambda}^*(t), \boldsymbol{\theta}^*(t))$ is a dual feasible point of $\hat{g}$, and it gives:

$$\begin{aligned} \hat{g}(\boldsymbol{\lambda}^*(t), \boldsymbol{\theta}^*(t)) &= \mathcal{J}(\boldsymbol{y}^*(t)) + \boldsymbol{\lambda}^*(t)^T \tilde{\boldsymbol{f}}(\boldsymbol{y}^*(t)) + \boldsymbol{\theta}^*(t)^T(\boldsymbol{A}\boldsymbol{y}^*(t) - \boldsymbol{b}) \\ &= \mathcal{J}(\boldsymbol{y}^*(t)) - n/t, \end{aligned} \tag{4.15}$$

where the second equality uses (4.11) and $\boldsymbol{\lambda}^*(t)^T \tilde{\boldsymbol{f}}(\boldsymbol{y}^*(t)) = \sum_{j=1}^N \sum_{i=1}^{n_j} \lambda_{i,j}^*(t) \tilde{f}_{i,j}(\boldsymbol{y}_j^*)$.

Since Slater's condition is satisfied by Assumption 2, the strong duality of problem (4.4) holds, namely

$$\max_{\boldsymbol{\lambda} \geq 0, \boldsymbol{\theta}} \hat{g}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \tilde{\mathcal{J}}^*. \tag{4.16}$$

By (4.15) and (4.16), it gives

$$\mathcal{J}(\boldsymbol{y}^*(t)) - \tilde{\mathcal{J}}^* \leq n/t. \tag{4.17}$$

That is to say, the worst approximation error of problem (4.8) w.r.t. problem (4.4) is $n/t$, which will be used in the next section to conceive a suboptimality criterion.

## 4.3 Primal dual interior point method

In this section, the primal dual interior point method introduced in Chapter 11 of [8] is fitted to problem setting of the previous section: (4.10) and (4.11) are interpreted as the modified KKT condition in solving problem (4.8), which is solved iteratively by Newton method.

### 4.3.1 Modified KKT condition

With KKT condition equations (4.10) and approximation analysis of §4.2.2, the modified KKT condition equations for problem (4.8) can be formulated as:

$$\boldsymbol{R}\boldsymbol{y} + \nabla\tilde{\boldsymbol{f}}(\boldsymbol{y})\boldsymbol{\lambda} + \boldsymbol{A}^T\boldsymbol{\theta} = \boldsymbol{0}, \tag{4.18a}$$

$$\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b} = \boldsymbol{0}, \tag{4.18b}$$

$$-\lambda_{i,j}\tilde{f}_{i,j}(\boldsymbol{y}_j) = 1/t,\ i = 1, ..., n_j,\ j = 1, ..., N. \tag{4.18c}$$

It can be concluded that a point $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ is primal and dual optimal of problem (4.8) if it satisfies (4.18) and $\boldsymbol{y}$ is in the domain of $\boldsymbol{\phi}$, namely $\tilde{\boldsymbol{f}}(\boldsymbol{y}) < \boldsymbol{0}$.

### 4.3.2 $\epsilon$ suboptimal solution via Newton method

Observing (4.17), it is intuitive that the higher $t$, the lower approximation bound of $n/t$, which can be finally decreased to 0 as $t$ approaches infinity.

By (4.15), the duality gap w.r.t. problem (4.4) for any point $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ satisfying the modified KKT condition (4.18) is $n/t$. Applying (4.11), this duality gap can be rewritten as $-\boldsymbol{\lambda}^T\tilde{\boldsymbol{f}}(\boldsymbol{y})$. When (4.18) is not satisfied by $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$, the notion of surrogate duality gap is introduced by the following definition to measure this intermediate approximation error.

**Definition 3.** *(Surrogate duality gap [8]) The surrogate duality gap w.r.t. problem* (4.4) *of a point* $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ *is said to be* $-\boldsymbol{\lambda}^T\tilde{\boldsymbol{f}}(\boldsymbol{y})$ *if and only if* $\boldsymbol{\lambda} \geq \boldsymbol{0}$ *and* $\tilde{\boldsymbol{f}}(\boldsymbol{y}) < \boldsymbol{0}$.

Incorporating Definition 3 and the modified KKT condition (4.18), the suboptimality of any point $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ w.r.t. problem (4.4) can be quantified by the definition below.

**Definition 4.** *($\epsilon$ suboptimal solution) Given $\epsilon > 0$, $\epsilon_p > 0$, and $\epsilon_d > 0$, a point $(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$, with $\boldsymbol{\lambda} > \boldsymbol{0}$ and $\tilde{\boldsymbol{f}}(\boldsymbol{y}) < \boldsymbol{0}$, is said to be an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution of problem* (4.4) *if it satisfies:*

$$||r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})|| \leq \epsilon_d,\ ||r_p(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})|| \leq \epsilon_p,\ -\boldsymbol{\lambda}^T\tilde{\boldsymbol{f}}(\boldsymbol{y}) \leq \epsilon,$$

*where*

$$r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \boldsymbol{R}\boldsymbol{y} + \nabla\tilde{\boldsymbol{f}}(\boldsymbol{y})\boldsymbol{\lambda} + \boldsymbol{A}^T\boldsymbol{\theta}, \tag{4.19}$$

$$r_p(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}. \tag{4.20}$$

To solve the modified KKT condition (4.18) as a whole, a Newton step $(\Delta\boldsymbol{y}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\theta})$ can be solved by the following linear equation:

$$KKT \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\theta} \end{bmatrix} = - \begin{bmatrix} r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta}) \\ r_s(\boldsymbol{y}, \boldsymbol{\lambda}) \\ r_p(\boldsymbol{y}) \end{bmatrix}, \tag{4.21}$$

where $r_s(\boldsymbol{y}, \boldsymbol{\lambda}) = -\operatorname{diag}(\boldsymbol{\lambda} D\tilde{\boldsymbol{f}}(\boldsymbol{y})) - (1/t) \cdot \mathbf{1}$, $\mathbf{1}$ is the vector of proper size with entries of 1, $D\tilde{\boldsymbol{f}}(\boldsymbol{y}) = (\nabla \bar{f}_{1,1}(\boldsymbol{y}_1), ..., \nabla \bar{f}_{n_1,1}(\boldsymbol{y}_1), ..., \nabla \bar{f}_{1,N}(\boldsymbol{y}_N), ..., \nabla \bar{f}_{n_N,N}(\boldsymbol{y}_N))^T$,

$$KKT = \begin{bmatrix} \boldsymbol{R} + D^2\tilde{\boldsymbol{f}}(\boldsymbol{y}) & D\tilde{\boldsymbol{f}}(\boldsymbol{y})^T & \boldsymbol{A}^T \\ -\operatorname{diag}(\boldsymbol{\lambda}) D\tilde{\boldsymbol{f}}(\boldsymbol{y}) & -\operatorname{diag}(\tilde{\boldsymbol{f}}(\boldsymbol{y})) & \mathbf{0} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

and $D^2\tilde{\boldsymbol{f}}(\boldsymbol{y}) = \operatorname{diag}(\lambda_{i,1} \sum_{i=1}^{n_1} \nabla^2 \tilde{f}_{i,1}(\boldsymbol{y}_1), ..., \sum_{i=1}^{n_N} \lambda_{i,N} \nabla^2 \tilde{f}_{i,N}(\boldsymbol{y}_N))$.

The iterative process in solving $(\boldsymbol{y}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\theta}^*(t))$ is in this manner: at the beginning of $(k+1)$-th iteration, the Newton step $(\Delta\boldsymbol{y}^k, \Delta\boldsymbol{\lambda}^k, \Delta\boldsymbol{\theta}^k)$ is solved by (4.21) using $(\boldsymbol{y}^k, \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k)$, and the $(k+1)$-th iteration is proceeded as:

$$\begin{bmatrix} \boldsymbol{y}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \\ \boldsymbol{\theta}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}^k \\ \boldsymbol{\lambda}^k \\ \boldsymbol{\theta}^k \end{bmatrix} + s \begin{bmatrix} \Delta\boldsymbol{y}^k \\ \Delta\boldsymbol{\lambda}^k \\ \Delta\boldsymbol{\theta}^k \end{bmatrix}, \tag{4.22}$$

where $s \in \mathbb{R}_{>0}$ is the step length, and will be determined by a back tracking line search in the PDIPM from Step 4 to 9 (Step 4 ensures the non negativeness of $\boldsymbol{\lambda}^{k+1}$).

To lighten the notation, let $r_d^k = r_d(\boldsymbol{y}^k, \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k)$, $r_p^k = r_p(\boldsymbol{y}^k)$, $r_s^k = r_s(\boldsymbol{y}^k, \boldsymbol{\lambda}^k)$, and $r^k = ((r_d^k)^T, (r_p^k)^T, (r_s^k)^T)^T$. The following algorithm PDIPM illuminates the primal dual interior point method to solve an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution of problem (4.4).

Note that, at each outer loop, parameter $t$ is increased by factor $\mu$ of $-n/(\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T\boldsymbol{\lambda}^k)$, which coincides with the intention that $t$ expands to infinity (see (4.17)) along the iteration as $\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T\boldsymbol{\lambda}^k$ approaches to 0 (regarding the optimality condition of problem (4.4)).

**Algorithm 6** Primal Dual Interior Point Method (PDIPM)

Initialize: $\tilde{\boldsymbol{f}}(\boldsymbol{y}^0) < \boldsymbol{0}$, $\boldsymbol{\lambda}^0 > \boldsymbol{0}$, $\boldsymbol{\theta}^0$, $\epsilon > 0$, $\epsilon_p > 0$, $\epsilon_d > 0$, $\mu > 1$, $\beta \in [0.3, 0.7]$, $\alpha \in [0.01, 0.1]$ and $k = 0$.
**repeat**
  $t = -\mu n/(\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T \boldsymbol{\lambda}^k)$
  $s = 0.99 \min\{1, \min\{-\lambda_{i,j}^k / \Delta\lambda_{i,j}^k \mid \Delta\lambda_{i,j}^k < 0\}\}$
  **repeat**
    get $(\Delta \boldsymbol{y}^k, \Delta \boldsymbol{\lambda}^k, \Delta \boldsymbol{\theta}^k)$ by (4.21)
    get $(\boldsymbol{y}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\theta}^{k+1})$ by (4.22)
    $s = \beta s$
  **until** $\tilde{\boldsymbol{f}}(\boldsymbol{y}^{k+1}) < \boldsymbol{0}$, and $||r^{k+1}|| \leq (1 - \alpha s)||r^k||$
  $k \leftarrow k + 1$
**until** $||r_d^k|| \leq \epsilon_d$, $||r_p^k|| \leq \epsilon_p$, and $-\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T \boldsymbol{\lambda}^k \leq \epsilon$
**return** $(\boldsymbol{y}^k, \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k)$

## 4.4 Accelerated termination of Primal dual interior point method for MPC

In MPC strategy, at each time instant, only the first step inputs of the control sequence, $\boldsymbol{u}_0$, would be applied to the system. With this in mind, the step separated structure of both objective function and constraints in problem (4.4) enables a step-based partition, which can be used to design an accelerated termination criterion in solving an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution when primal dual interior point method is implemented.

### 4.4.1 Accelerated termination criterion

At current time instant, only inputs of the first step in the prediction horizon are applied in MPC. At subsequent time instant, the initial state $\bar{x}_l$ will be updated, based on which the renewed MPC problem (4.4) with the exact prediction horizon length $N$ is to be solved. As a consequence, the first step decision variables are sufficient for MPC control law. And the first step focused criterion is derived in this subsection.

As a prerequisite, $\boldsymbol{A}$, $\boldsymbol{b}$ and $\tilde{\boldsymbol{f}}(\boldsymbol{y})$ can be decomposed into block form as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{(1:1)} & \boldsymbol{0} \\ \boldsymbol{B}_{(2:N)} & \boldsymbol{A}_{(2:N)} \end{bmatrix}, \ \tilde{\boldsymbol{f}}(\boldsymbol{y}) = \begin{bmatrix} \tilde{\boldsymbol{f}}_{(1:1)}(\boldsymbol{y}_1) \\ \tilde{\boldsymbol{f}}_{(2:N)}(\boldsymbol{y}_{(2:N)}) \end{bmatrix},$$

where $\boldsymbol{A}_{(1:1)} = B_1$, $\boldsymbol{A}_{(1:1)} \in \mathbb{R}^{nx \times ny}$, $\boldsymbol{B}_{(2:N)} = (A_2^T, \mathbf{0}, ..., \mathbf{0})^T$, $\boldsymbol{B}_{(2:N)} \in \mathbb{R}^{(N-1)nx \times ny}$ ,

$$\boldsymbol{A}_{(2:N)} = \begin{bmatrix} B_2 & & & \\ A_3 & B_3 & & \\ & \ddots & \ddots & \\ & & A_N & B_N \end{bmatrix},$$

$A_{(2:N)} \in \mathbb{R}^{(N-1)nx \times (N-1)ny}$, $\tilde{\boldsymbol{f}}_{(2:N)}(\boldsymbol{y}_{(2:N)}) = (\tilde{\boldsymbol{f}}_2(\boldsymbol{y}_2)^T, ..., \tilde{\boldsymbol{f}}_N(\boldsymbol{y}_N)^T)^T$, and for $j = 1, ..., N$, the expression of $A_j$ and $B_j$ can be referred to (1.6).

As a result, equality constraint (4.4b) can be partitioned as:

$$\begin{aligned} &\boldsymbol{A}_{(1:1)}\boldsymbol{y}_1 = \boldsymbol{b}_{(1:1)}, \\ &\boldsymbol{B}_{(2:N)}\boldsymbol{y}_1 + A_{(2:N)}\boldsymbol{y}_{(2:N)} = \boldsymbol{b}_{(2:N)}. \end{aligned}$$

Likewise, (4.19) can be partitioned as

$$\begin{aligned} r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta}) &= \begin{bmatrix} r_{d,(1:1)}(\boldsymbol{y}_1, \boldsymbol{\lambda}_1, \boldsymbol{\theta}_1, \boldsymbol{\theta}_{(2:N)}) \\ r_{d,(2:N)}(\boldsymbol{y}_{(2:N)}, \boldsymbol{\lambda}_{(2:N)}, \boldsymbol{\theta}_{(2:N)}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_1\boldsymbol{y}_1 + \nabla\tilde{\boldsymbol{f}}_1(\boldsymbol{y}_1)\boldsymbol{\lambda}_1 + \boldsymbol{A}_{(1:1)}^T\boldsymbol{\theta}_1 + \boldsymbol{B}_{(2:N)}^T\boldsymbol{\theta}_{(2:N)} \\ \boldsymbol{R}_{(2:N)}\boldsymbol{y}_{(2:N)} + \nabla\tilde{\boldsymbol{f}}_{(2:N)}(\boldsymbol{y}_{(2:N)})\boldsymbol{\lambda}_{(2:N)} + \boldsymbol{A}_{(2:N)}^T\boldsymbol{\theta}_{(2:N)}. \end{bmatrix} \end{aligned}$$

In the continuation, the following problem is considered.

$$\min_{\boldsymbol{y}_{(2:N)}} \ \mathcal{J}_{(2:N)}(\boldsymbol{y}_{(2:N)}), \tag{4.24a}$$

$$s.t. \ \boldsymbol{B}_{(2:N)}\bar{\boldsymbol{y}}_1 + \boldsymbol{A}_{(2:N)}\boldsymbol{y}_{(2:N)} = \boldsymbol{b}_{(2:N)}, \tag{4.24b}$$

$$\tilde{\boldsymbol{f}}_{(2:N)}(\boldsymbol{y}_{(2:N)}) < \mathbf{0}, \tag{4.24c}$$

where $\mathcal{J}_{(2:N)}(\boldsymbol{y}_{(2:N)}) = \frac{1}{2}||\boldsymbol{y}_{(2:N)}||^2_{\boldsymbol{R}_{(2:N)}}$.

Suppose that there exists $\bar{\boldsymbol{y}}_1$ enabling problem (4.24) to be solvable, then there must exist $(\hat{\boldsymbol{y}}_{(2:N)}, \hat{\boldsymbol{\lambda}}_{(2:N)}, \hat{\boldsymbol{\theta}}_{(2:N)})$ satisfying the corresponding KKT conditions as follows.

$$\boldsymbol{R}_{(2:N)}\hat{\boldsymbol{y}}_{(2:N)} + \boldsymbol{A}_{(2:N)}^T\hat{\boldsymbol{\theta}}_{(2:N)} + \nabla\tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)})\hat{\boldsymbol{\lambda}}_{(2:N)} = \mathbf{0}, \tag{4.25a}$$

$$\hat{\boldsymbol{\lambda}}_{(2:N)}\tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)}) = \mathbf{0}, \tag{4.25b}$$

$$\hat{\boldsymbol{\lambda}}_{(2:N)} \geq \mathbf{0}, \tag{4.25c}$$

$$\boldsymbol{B}_{(2:N)}\bar{\boldsymbol{y}}_1 + \boldsymbol{A}_{(2:N)}\hat{\boldsymbol{y}}_{(2:N)} = \boldsymbol{b}_{(2:N)}. \tag{4.25d}$$

Now, given $\bar{\boldsymbol{y}}_1 \in \mathbb{R}^{nx+nu}$, $\bar{\boldsymbol{\lambda}}_1 \in \mathbb{R}^{n_1}_{>0}$, and $\bar{\boldsymbol{\theta}}_1 \in \mathbb{R}^{nx+na_1}$, the norm minimization problem below is considered:

$$\min_{\boldsymbol{\theta}_{(2:N)}} \; ||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \boldsymbol{\theta}_{(2:N)})||. \tag{4.26}$$

The following assumption will help to design the first step focused criterion for an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution of problem (4.4).

**Assumption 4.** *Let $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}})$ be generated by PDIPM, $\epsilon$, $\epsilon_p$ and $\epsilon_d$ are assumed to be sufficiently small such that when*

$$||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_{(2:N)})|| < \epsilon_d, \tag{4.27}$$

$$||\boldsymbol{A}_{(1:1)}\bar{\boldsymbol{y}}_1 - \boldsymbol{b}_{(1:1)}|| \leq \epsilon_p, \tag{4.28}$$

$$-\bar{\lambda}_1 \tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{y}}_1) < \epsilon, \tag{4.29}$$

*are satisfied, it leads to:*

$$||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_{(2:N)})|| \leq ||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_{(2:N)})||. \tag{4.30}$$

**Remark 4.** *Denote $\boldsymbol{y}^*$ the optimal solution of problem* (4.4)*, and denote its associated dual variable of* (4.4b) *as $\boldsymbol{\theta}^*$. In implementing PDIPM, it is evident that as $\bar{\boldsymbol{y}}_1$ converges to $\boldsymbol{y}_1^*$, both $\bar{\boldsymbol{\theta}}_{(2:N)}$ and $\hat{\boldsymbol{\theta}}_{(2:N)}$ converge to $\boldsymbol{\theta}^*_{(2:N)}$. And when $\bar{\boldsymbol{y}}_1 = \boldsymbol{y}_1^*$, it must have $\hat{\boldsymbol{\theta}}_{(2:N)} = \boldsymbol{\theta}^*_{(2:N)}$, but not necessarily $\bar{\boldsymbol{\theta}}_{(2:N)} = \boldsymbol{\theta}^*_{(2:N)}$. As such, when $(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1)$ is close enough to $(\boldsymbol{y}_1^*, \boldsymbol{\lambda}_1^*, \boldsymbol{\theta}_1^*)$, it is acceptable to assume that $\hat{\boldsymbol{\theta}}_{(2:N)}$ is closer than $\bar{\boldsymbol{\theta}}_{(2:N)}$ to $\boldsymbol{\theta}^*_{(2:N)}$ in terms of Euclidean norm. Since problem* (4.26) *is unconstrained convex optimization,* (4.30) *should hold when $\epsilon$, $\epsilon_p$ and $\epsilon_d$ are sufficiently small and* (4.27) (4.28) (4.29) *are satisfied.*

**Remark 5.** *Note that when inequality constraints are linear, problem* (4.4) *becomes quadratic linear constrained optimization. Since Newton step solved in* (4.21) *can be interpreted as solving an equation: making gradient of quadratic approximation of expression to be minimized being 0. Recall* (4.19) *and* (4.13)*, $r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ is indeed the gradient of $\hat{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})$, resulting $||r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})||$ reduces to extreme good value, say equal or less than magnitude of $10^{-12}$ with just few iterations (showed in Table 4.2.). To conclude, Assumption 4 is quite reasonable when* (4.4c) *is linear.*

**Proposition 1.** *Let* $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}})$ *be generated by PDIPM, if* (4.27) (4.28), *and* (4.29) *are satisfied, and* $\bar{\boldsymbol{y}}_1$ *enables problem* (4.24) *to be solvable, then* $(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1)$ *is the first step components of an* $(\epsilon, \epsilon_p, \epsilon_d)$ *suboptimal solution of problem* (4.4).

*Proof.* By (4.25b) (4.25c), it gives $\hat{\boldsymbol{\lambda}}_{(2:N)} = \mathbf{0}$. Since $\tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)}) < \mathbf{0}$, there must exist a sufficiently small $\boldsymbol{\lambda}'_{(2:N)} \in \mathbb{R}^{n_2+\ldots+n_N}_{>0}$ such that

$$||\nabla \tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)})\boldsymbol{\lambda}'_{(2:N)}||^2 \leq \epsilon_d^2 - ||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_{(2:N)})||^2, \tag{4.31}$$

$$-\boldsymbol{\lambda}'_{(2:N)}\tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)}) \leq \epsilon + \bar{\lambda}_1 \tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{y}}_1), \tag{4.32}$$

Finally, it can be concluded that:

$$\begin{aligned} &\left\| \boldsymbol{A} \begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_{(2:N)} \end{bmatrix} - \boldsymbol{b} \right\| = \left\| \begin{bmatrix} \boldsymbol{A}_{(1:1)}\bar{\boldsymbol{y}}_1 - \boldsymbol{b}_{(1:1)} \\ \boldsymbol{B}_{(2:N)}\bar{\boldsymbol{y}}_1 + \boldsymbol{A}_{(2:N)}\hat{\boldsymbol{y}}_{(2:N)} - \boldsymbol{b}_{(2:N)} \end{bmatrix} \right\| \\ &= ||\boldsymbol{A}_{(1:1)}\bar{\boldsymbol{y}}_1 - \boldsymbol{b}_{(1:1)}|| \leq \epsilon_p, \end{aligned} \tag{4.33}$$

$$\begin{aligned} &\left\| \boldsymbol{R} \begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_{(2:N)} \end{bmatrix} + \nabla \tilde{\boldsymbol{f}} \left( \begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_{(2:N)} \end{bmatrix} \right) \begin{bmatrix} \bar{\boldsymbol{\lambda}}_1 \\ \boldsymbol{\lambda}'_{(2:N)} \end{bmatrix} + A^T \begin{bmatrix} \bar{\boldsymbol{\theta}}_1 \\ \hat{\boldsymbol{\theta}}_{(2:N)} \end{bmatrix} \right\| \\ &\leq \left\| \begin{bmatrix} r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_{(2:N)}) \\ \nabla \tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)})\boldsymbol{\lambda}'_{(2:N)} \end{bmatrix} \right\| \leq \epsilon_d, \end{aligned} \tag{4.34}$$

$$-\begin{bmatrix} \bar{\boldsymbol{\lambda}}_1 \\ \boldsymbol{\lambda}'_{(2:N)} \end{bmatrix}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_1(\bar{\boldsymbol{y}}_1) \\ \tilde{\boldsymbol{f}}_{(2:N)}(\hat{\boldsymbol{y}}_{(2:N)}) \end{bmatrix} \leq \epsilon, \tag{4.35}$$

where the inequality of (4.33) uses (4.25d), the first and second inequalities of (4.34) use (4.30) and (4.31) respectively, and the inequality of (4.35) uses (4.32).

Consequently, by Definition 4, $\left( \begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_{(2:N)} \end{bmatrix}, \begin{bmatrix} \bar{\boldsymbol{\lambda}}_1 \\ \boldsymbol{\lambda}'_{(2:N)} \end{bmatrix}, \begin{bmatrix} \bar{\boldsymbol{\theta}}_1 \\ \hat{\boldsymbol{\theta}}_{(2:N)} \end{bmatrix} \right)$ is an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution of problem (4.4). And $(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1)$ is the first step components of an $(\epsilon, \epsilon_p, \epsilon_d)$ suboptimal solution of problem (4.4). □

### 4.4.2 The accelerated termination algorithm

In this subsection, an accelerated termination algorithm of primal dual interior point method for MPC is delineated, which employs Proposition 1 and gives the proof of fewer iterations required for solving a suboptimal solution of the same level.

First, problem (4.36) can be used to test whether problem (4.24) is solvable, which indeed verifies if the intersection of (4.24b) and (4.24c) is empty or not.

$$\min_{\boldsymbol{y}_{(2:N)},h} h \tag{4.36a}$$

$$s.t.\ \tilde{\boldsymbol{f}}_{(2:N)}(\boldsymbol{y}_{(2:N)}) < he, \tag{4.36b}$$

$$\boldsymbol{B}_{(2:N)}\bar{\boldsymbol{y}}_1 + \boldsymbol{A}_{(2:N)}\boldsymbol{y}_{(2:N)} = \boldsymbol{b}_{(2:N)}. \tag{4.36c}$$

$$h \leq 0, \tag{4.36d}$$

where $e$ is column vector of proper size with entries of 1.

**Algorithm 7** Accelerated Termination of Primal Dual Interior Point Method(ATPDIPM)

Initialize: $\tilde{\boldsymbol{f}}(\boldsymbol{y}^0) < \boldsymbol{0}$, $\boldsymbol{\lambda}^0 > \boldsymbol{0}$, $\boldsymbol{\theta}^0$, $\epsilon > 0$, $\epsilon_p > 0$, $\epsilon_d > 0$, $\mu > 1$, $\beta \in [0.3, 0.7]$, $\alpha \in [0.01, 0.1]$ and $k = 0$.
**while** 1 **do**
  $t = -\mu n/(\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T\boldsymbol{\lambda}^k)$
  $s = 0.99\min\{1, \min\{-\lambda^k_{i,j}/\Delta\lambda^k_{i,j} \mid \Delta\lambda^k_{i,j} < 0\}\}$
  **repeat**
    get $(\Delta\boldsymbol{y}^k, \Delta\boldsymbol{\lambda}^k, \Delta\boldsymbol{\theta}^k)$ by (4.21)
    get $(\boldsymbol{y}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\theta}^{k+1})$ by (4.22)
    $s = \beta s$
  **until** $\tilde{\boldsymbol{f}}(\boldsymbol{y}^{k+1}) < \boldsymbol{0}$, and $||r^{k+1}|| \leq (1 - \alpha s)||r^k||$
  **if** (4.27) (4.28) (4.29) are sasisfied **then**
    **if** problem (4.36) has a solution by taking $\boldsymbol{y}_1^{k+1}$ as $\bar{\boldsymbol{y}}_1$ **then Break**
    **else if** $||r_d^{k+1}|| \leq \epsilon_d$, $||r_p^{k+1}|| \leq \epsilon_p$, and $-\tilde{\boldsymbol{f}}(\boldsymbol{y}^{k+1})^T\boldsymbol{\lambda}^{k+1} \leq \epsilon$ **then Break**
    **end if**
  **end if**
  $k \leftarrow k + 1$
**end while**
**return** $(\boldsymbol{y}_1^{k+1}, \boldsymbol{\lambda}_1^{k+1})$

**Proposition 2.** *Given $\epsilon > 0$, $\epsilon_p > 0$, and $\epsilon_d > 0$, termination of ATPDIPM requires fewer or the same iterations than that of PDIPM.*

*Proof.* At $k$-th iteration, after each inner loop of PDIPM and ATPDIPM, it gives $\forall j = 1, ..., N$, $\tilde{\boldsymbol{f}}_j(\boldsymbol{y}_j^{k+1}) < \boldsymbol{0}$ and $\boldsymbol{\lambda}_j^{k+1} > 0$, thus $-\tilde{\boldsymbol{f}}_j(\boldsymbol{y}_j^{k+1})^T\boldsymbol{\lambda}_j^{k+1} > 0$. In accordance, if $-\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T\boldsymbol{\lambda}^k \leq \epsilon$, it must give $-\tilde{\boldsymbol{f}}_1(\boldsymbol{y}_1^{k+1})^T\boldsymbol{\lambda}_1^{k+1} \leq \epsilon$.

Suppose that $||r_d^k|| \leq \epsilon_d$, $||r_p^k|| \leq \epsilon_p$, and $-\tilde{\boldsymbol{f}}(\boldsymbol{y}^k)^T\boldsymbol{\lambda}^k \leq \epsilon$ are satisfied at $k$-th iteration in implementation of PDIPM, they must also be satisfied at $k$-th iteration in implementation of ATPDIPM, since the same iterative process (4.22) is utilized. However, if prior to $k$-th iteration, problem (4.36) has a solution, then ATPDIPM can be terminated. And this completes the proof. □

## 4.5 Numerical experiments

In this section, in total 500 randomly generated tests (100 tests per each prediction horizon from 5 to 25 with interval of 5) are carried out to compare the general performance between the conventional and the first step focused primal dual interior point method. Each test are implemented under 3 different relative suboptimality [1]: $1 \times 10^{-3}$, $1 \times 10^{-4}$, $1 \times 10^{-5}$. For each test, the parameter for stopping condition are: $\epsilon_d = 1 \times 10^{-10}$, and $\epsilon_p = 1 \times 10^{-4}$. All numerical experiments are carried out using MATLAB 2020b on a Windows 10 PC with 2.20 GHz Core i7-8750H CPU and 16GB RAM.

In detail, the tested system consists of 5 subsystems, and $\forall l = 1, ..., 5$, $i = 1, ..., 5$, $nx_l = nu_l = 2$, $A_{li}$ and $B_{li}$ are randomly generated by MATLAB command `drss`, inequality constraints (4.4c) are as: $0 \leq \boldsymbol{u} \leq 1$. The penalty matrix is set as $\boldsymbol{R} = \boldsymbol{I}$. Each element of $\bar{x}_l$ is drawn from uniform distribution $[-0.5, 0.5]$. For both PDIPM and ATPDIPM, parameters for back tracking line search are: $\alpha = 0.1$, $\beta = 0.5$ and $\mu = 10$, parameters for initialization are: $\boldsymbol{y}^0 = 0.5 \cdot \boldsymbol{1}$, $\boldsymbol{\lambda}^0 = \boldsymbol{1}$, and $\boldsymbol{\theta}^0 = 0 \cdot \boldsymbol{1}$.

Note that results of ATPDIPM in terms of $\epsilon$, $\epsilon_p$ and $\epsilon_d$ are computed from $(\boldsymbol{y}_1^{k+1} \oplus \hat{\boldsymbol{y}}_{(2:N)}, \boldsymbol{\lambda}_1^{k+1} \oplus \hat{\boldsymbol{\lambda}}_{(2:N)}, \boldsymbol{\theta}_1^{k+1} \oplus \hat{\boldsymbol{\theta}}_{(2:N)})$, where $(\boldsymbol{y}_1^{k+1}, \boldsymbol{\lambda}_1^{k+1}, \boldsymbol{\theta}_1^{k+1})$ are first step components of $(\boldsymbol{y}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\theta}^{k+1})$. Note that in Proposition 1, a sufficiently small $\boldsymbol{\lambda}'_{(2:N)}$ intervenes in the full sequence solution, but it is most for theoretical proof concern, and has no effect to results in numerical experiments if $\boldsymbol{\lambda}'_{(2:N)}$ takes infinitely small positive value.

1. The relative suboptimality is computed as suboptimality divided by $\tilde{\mathcal{J}}^*$.

Table 4.1 – Overall performance comparison between PDIPM and ATPDIPM of one single test with $N = 15$

| Predefined relative $\epsilon$ | $1 \times 10^{-5}$ | | $1 \times 10^{-4}$ | | $1 \times 10^{-3}$ | |
|---|---|---|---|---|---|---|
| Algorithm | PDIPM | ATPDIPM | PDIPM | ATPDIPM | PDIPM | ATPDIPM |
| Iteration # | 22 | 19 | 21 | 17 | 19 | 16 |
| Time (s) | 0.5727 | 0.4781 | 0.5067 | 0.4169 | 0.4405 | 0.3935 |
| Relative $\epsilon$ | 2.37e-7 | 1.02e-6 | 2.21e-6 | 2.58e-5 | 4.94e-5 | 1.31e-4 |
| $\|\|r_p\|\|$ | 3.53e-16 | 2.48e-9 | 3.37e-16 | 2.46e-9 | 3.89e-16 | 6.85e-9 |
| $\|\|r_d\|\|$ | 1.90e-15 | 4.71e-16 | 1.91e-15 | 5.36e-16 | 2.54e-15 | 4.25e-16 |

### 4.5.1 Specific performance of a single test

To let the reader catch a glimpse of the exact indicators of PDIPM and ATPDIPM, their main performance is detailed by a single test with $N = 15$. Please refer to Appendix A for parameters setting of this single test.

Table 4.1 shows that the predefined relative $\epsilon$, $\epsilon_p$ and $\epsilon_d$ are fulfilled by 2 algorithms, of which PDIPM preserves higher precision in terms of $\epsilon$ and $\epsilon_p$. Of all predefined relative $\epsilon$, ATPDIPM consumes 3-4 fewer iterations than that of PDIPM, resulting in a proportional advantage on computation time.

### 4.5.2 Statistics comparison among 500 tests

In this subsection, the statistics of 500 random tests are exhibited. As results of relative $\epsilon$, $\epsilon_p$ and $\epsilon_d$ reveal no noticeable statistical discrepancies among different prediction horizons, the results of each prediction horizons are therefore aggregated into one item, as showed in Table 4.2.

Statistics in Table 4.2 exhibit that the predefined relative $\epsilon$, $\epsilon_p$ and $\epsilon_d$ are fulfilled by 2 algorithms. PDIPM preserves higher precision in terms of $\epsilon$ and $\epsilon_p$, because in ATPDIPM, suboptimality and primal residual are mainly caused by the first step, which usually are just below the predefined bound. As for PDIPM, it generally takes 2-4 more iterations to terminate, and can achieve much higher precision for full sequence due to the quadratic convergence rate. In terms of $\epsilon_d$, as explained in Remark 5 that $||r_d(\boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\theta})||$ reduces to extreme good value in just few iterations, which makes $||r_{d,(1:1)}(\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_{(2:N)})||$ becoming smaller after swapping $\bar{\boldsymbol{\theta}}_{(2:N)}$ for $\hat{\boldsymbol{\theta}}_{(2:N)}$. Given $||r_{d,(2:N)}(\hat{\boldsymbol{y}}_{(2:N)}, \hat{\boldsymbol{\lambda}}_{(2:N)}, \bar{\boldsymbol{\theta}}_{(2:N)}||$ can take 0 when quit happens in ATPDIPM, the overall $||r_d||$ can be less than that of PDIPM.

Table 4.2 – Statistics of suboptimality between PDIPM and ATPDIPM of 500 tests

| Predefined relative $\epsilon$ | $1 \times 10^{-5}$ | | $1 \times 10^{-4}$ | | $1 \times 10^{-3}$ | |
|---|---|---|---|---|---|---|
| Algorithm | PDIPM | ATPDIPM | PDIPM | ATPDIPM | PDIPM | ATPDIPM |
| Ave. Rel. $\epsilon$ | 4.22e-7 | 1.29e-6 | 6.01e-6 | 1.56e-5 | 6.7e-5 | 1.67e-4 |
| Max. Rel. $\epsilon$ | 2.18e-6 | 4.53e-6 | 2.36e-5 | 7.3e-5 | 2.89e-4 | 5.77e-4 |
| Ave. $\epsilon_p$ | 1.55e-13 | 8.11e-9 | 4.68e-11 | 1.51e-7 | 1.62e-9 | 3.07e-5 |
| Max. $\epsilon_p$ | 2.95e-11 | 2.98e-7 | 1.73e-8 | 2.95e-6 | 2.95e-7 | 9.8e-5 |
| Ave. $\epsilon_d$ | 1.19e-14 | 4.02e-15 | 1.14e-14 | 3.15e-15 | 1.05e-14 | 3.02e-15 |
| Max. $\epsilon_d$ | 2.52e-12 | 9.19e-13 | 2.53e-12 | 6.17e-13 | 2.24e-12 | 7.45e-13 |

From Fig. 4.1, Fig. 4.3 and Fig. 4.5, ATPDIPM consumes less iterations than PDIPM under all predefined relative $\epsilon$, which coincides with Proposition 1. In terms of the impact on iteration number ratio by predefined suboptimality, no significant differences are shown among these 3 figures, indicating that the iteration number needed for PDIPM and ATPDIPM changes proportionally as predefined suboptimality varies. Indeed, both methods spend a few more iterations steadily as predefined suboptimality decreases tenfold, which verified the behaviors of iteration number resulting from quadratic convergence rate.

From Fig. 4.2, Fig. 4.4 and Fig. 4.6, ATPDIPM spends statistically less time than PDIPM under all predefined relative $\epsilon$ for prediction horizon of 15, 20, and 25. For $N = 10$, the majority cases of ATPDIPM possess an advantage on computation time over PDIPM. However, for $N = 5$, due to an extremely short prediction horizon, the gain by fewer iterations is not enough to cover loss by a more complex quit mechanism, leading to overall inferiority of ATPDIPM in computation time.

## 4.6 Conclusion

In this chapter, focusing on first step components, an accelerated termination criterion has been proposed to generated $(\epsilon, \epsilon_p, \epsilon_d)$ solution of the general MPC resulted convex optimization problem in implementation of the primal dual interior point method.

The effectiveness of the accelerated termination criterion has been demonstrated under the mild assumption. In terms of iteration number, the superiority of the new stopping criterion based algorithm has been demonstrated by mathematical proof and random numerical experiments. Concerning computation time, the dominance of the newly proposed algorithm has been testified for medium to long prediction horizon.

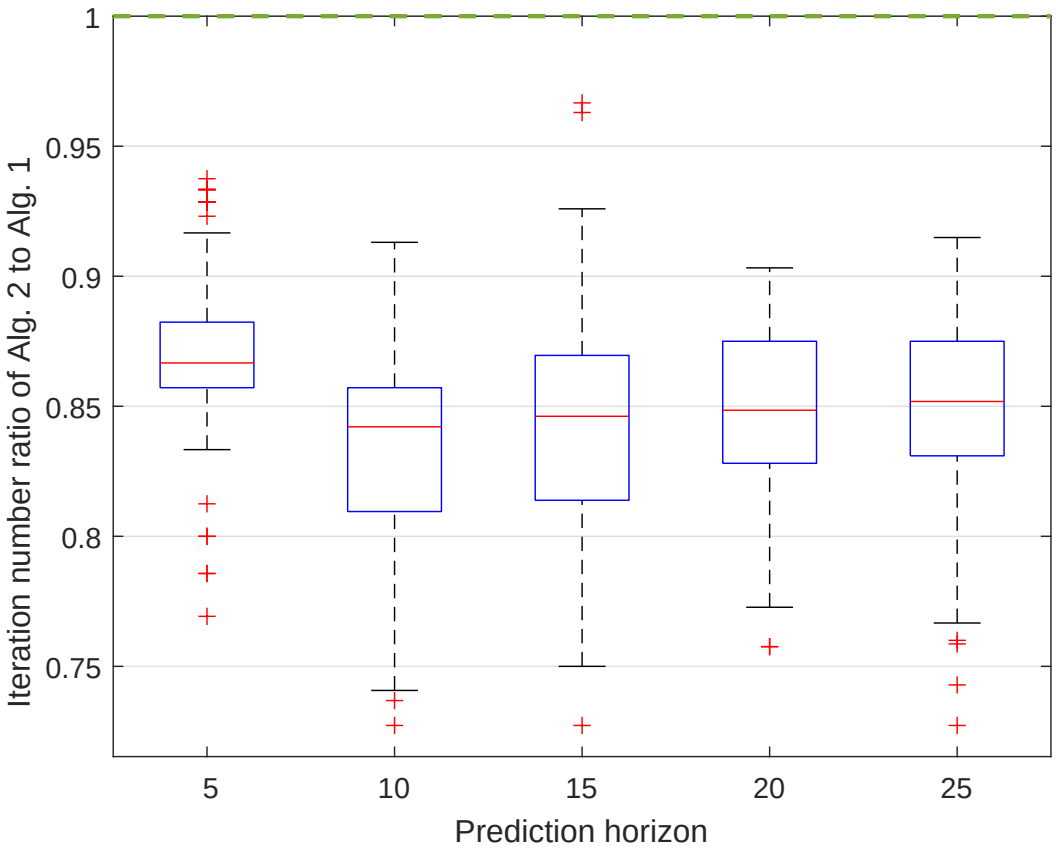


Figure 4.1 – Iteration number ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-5}$
Sample value exceeded $+/-2.7\sigma$ shows as whisker, the same setting for other box plots. Sample value less, greater than, or equals to 1 (green horizontal line) means ATPDIPM consumes fewer, more, or the same iterations as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

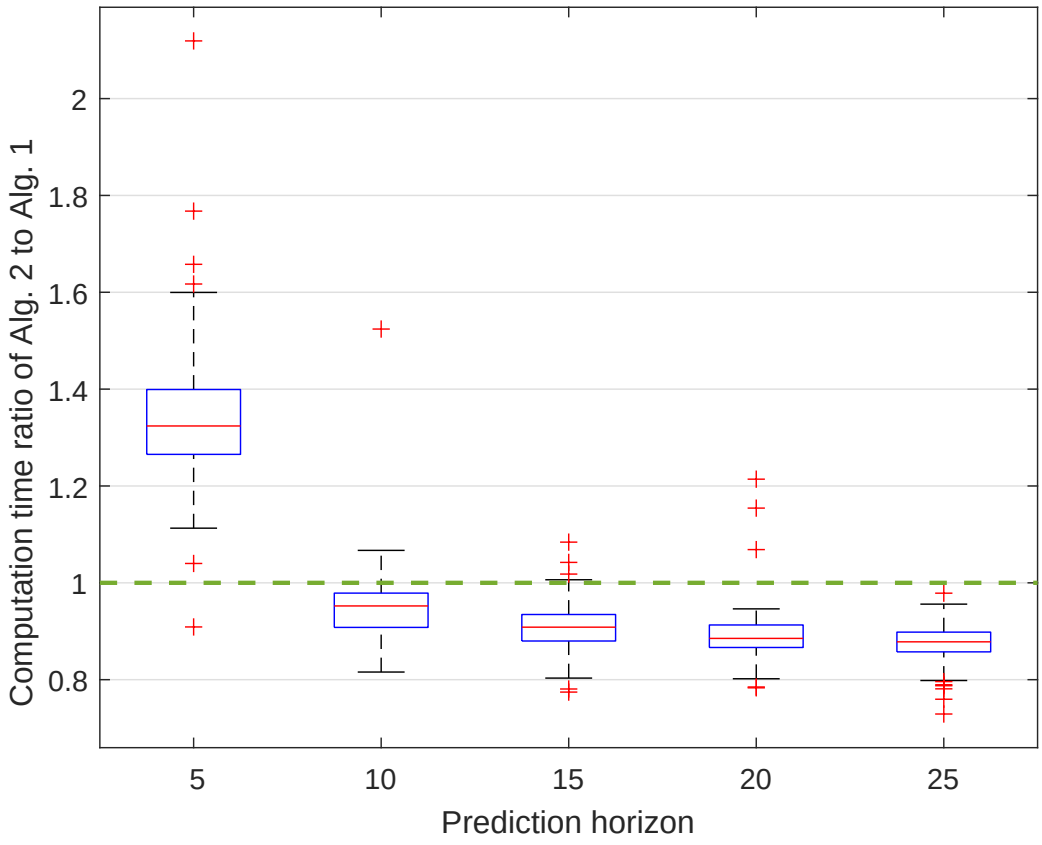


Figure 4.2 – Computation time ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-5}$
Sample value less, greater than or equals to 1 (green horizontal line) means ATPDIPM spends less, more or the same time as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

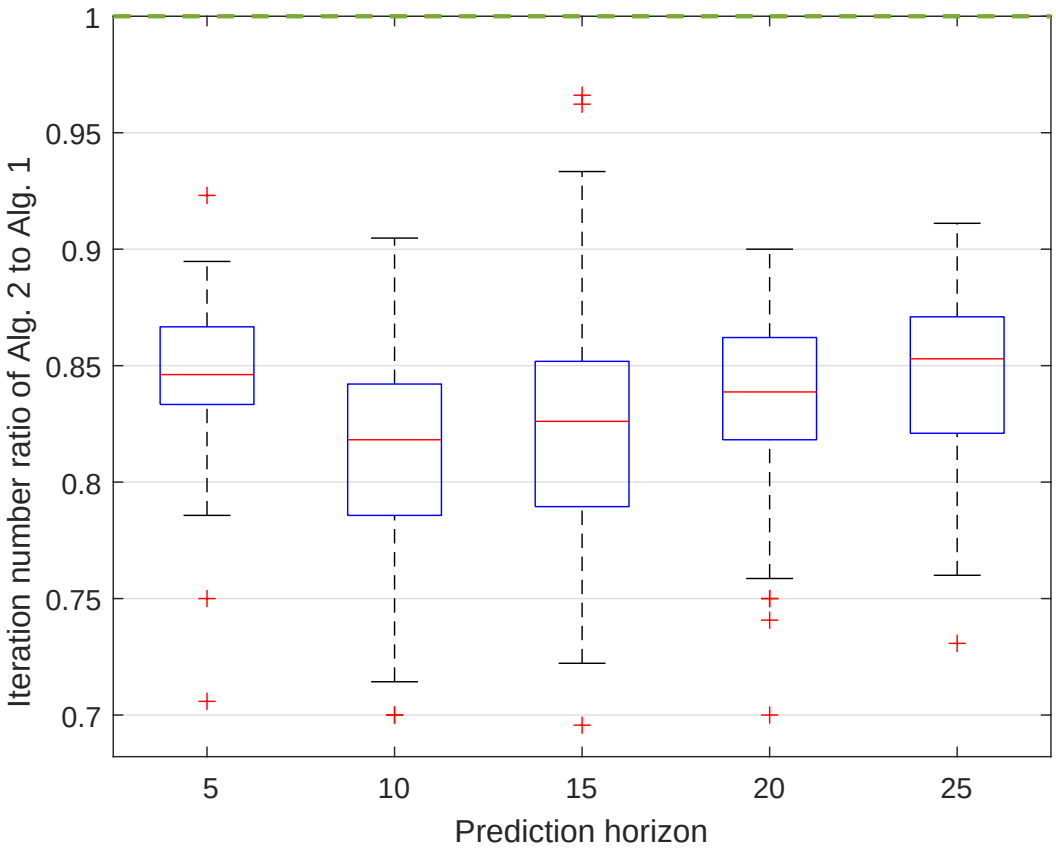


Figure 4.3 – Iteration number ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-5}$
Sample value less, greater than, or equals to 1 (green horizontal line) means ATPDIPM consumes fewer, more, or the same iterations as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

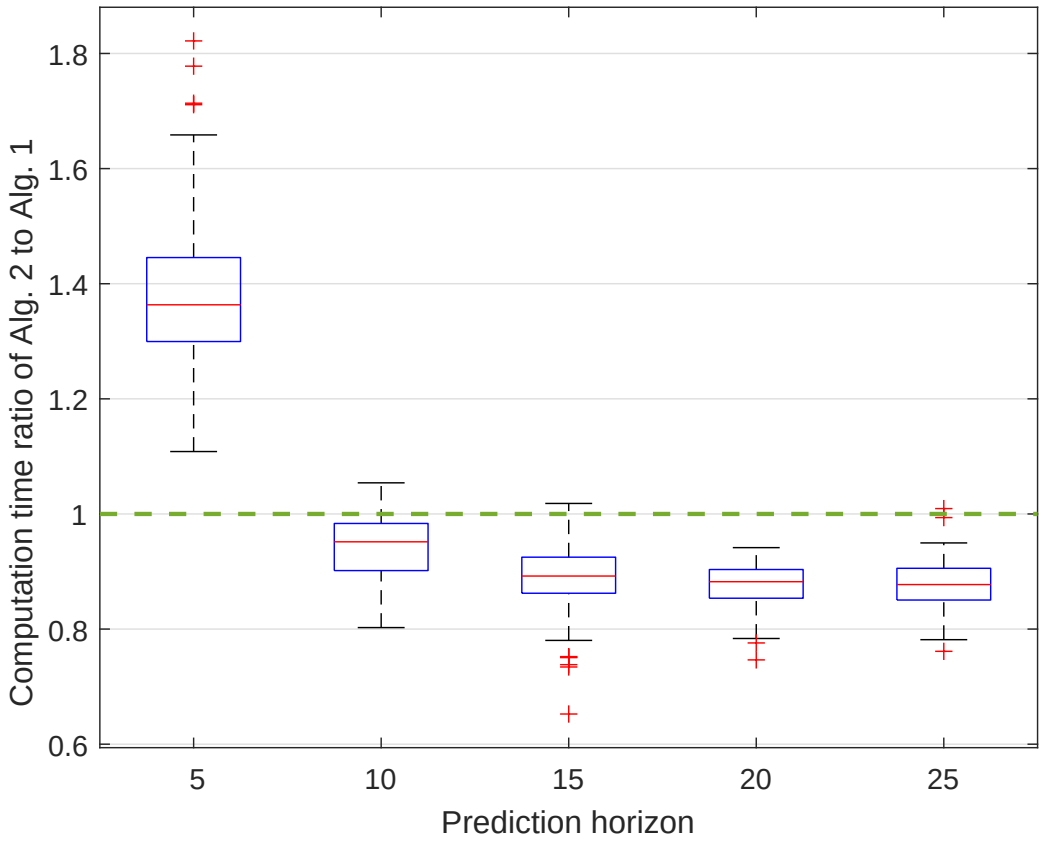


Figure 4.4 – Computation time ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-4}$
Sample value less, greater than or equals to 1 (green horizontal line) means ATPDIPM spends less, more or the same time as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

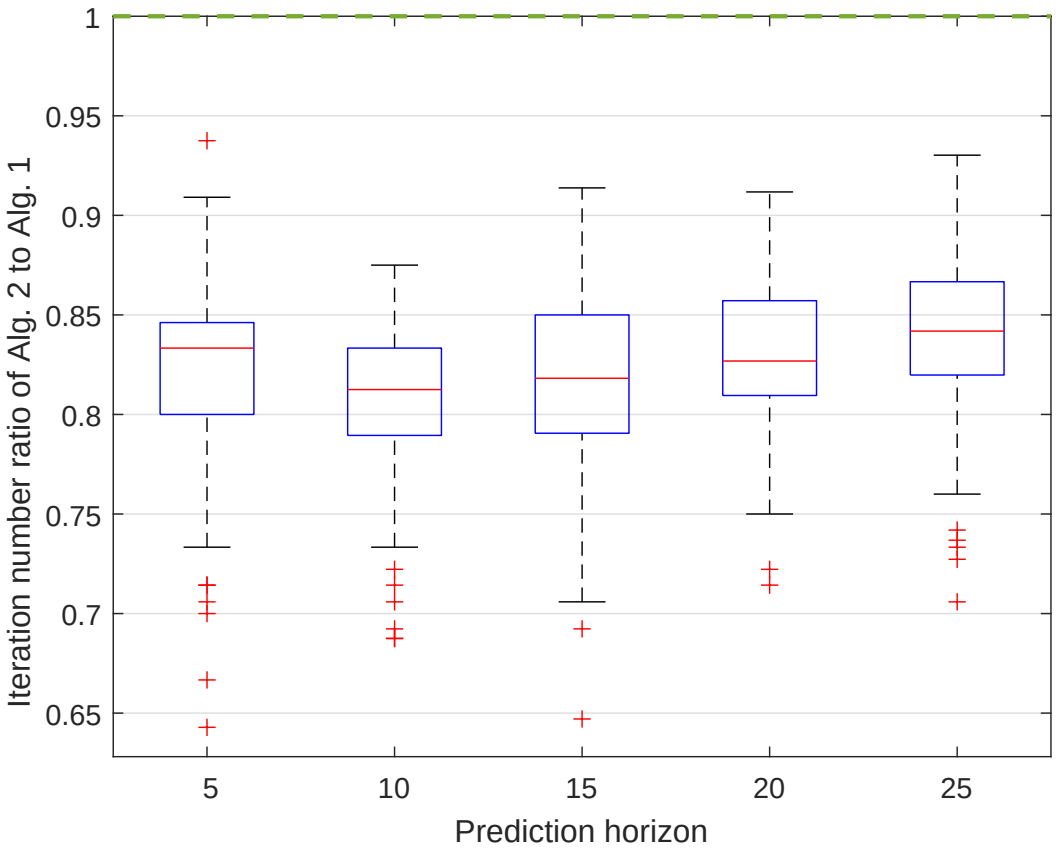


Figure 4.5 – Iteration number ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-3}$

Sample value less, greater than, or equals to 1 (green horizontal line) means ATPDIPM consumes fewer, more, or the same iterations as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

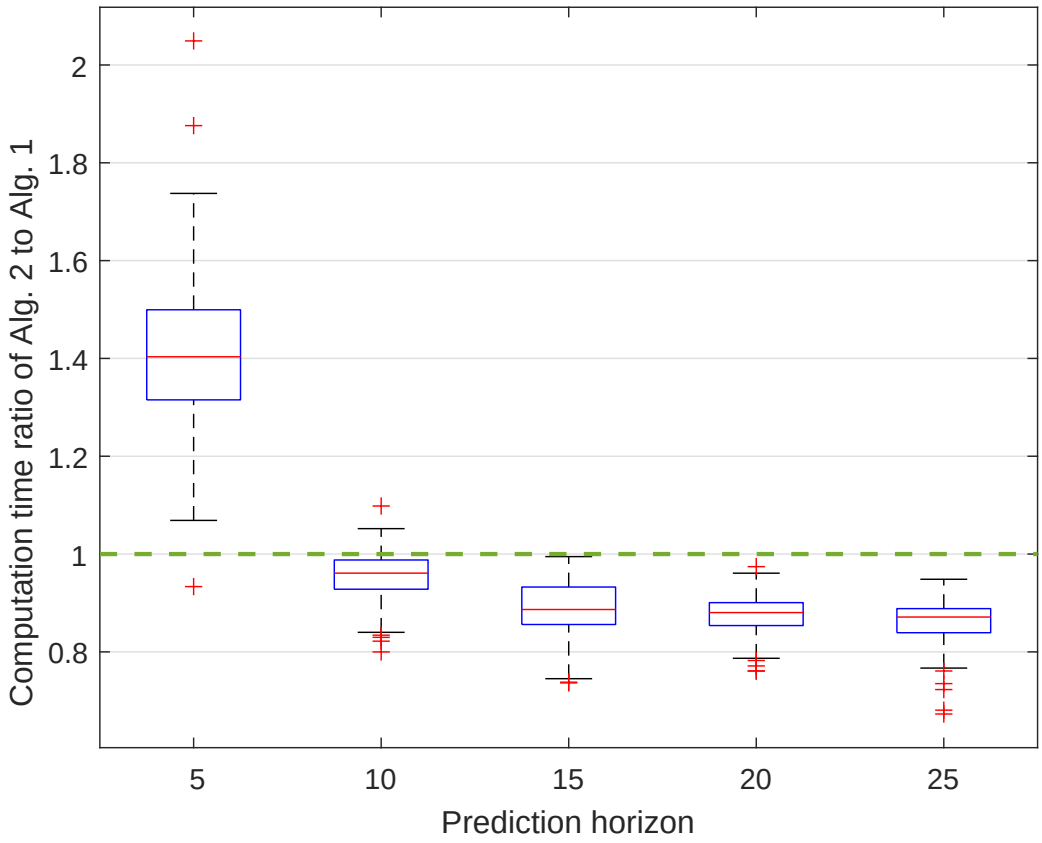


Figure 4.6 – Computation time ratio of ATPDIPM to PDIPM with predefined relative $\epsilon$ as $1 \times 10^{-3}$
Sample value less, greater than or equals to 1 (green horizontal line) means ATPDIPM spends less, more or the same time as PDIPM in the same test. The lower value, the better performance of ATPDIPM.

In §5, taking a step out of the MPC domain, the general linear constrained quadratic optimization will be handled. In contrast to this section, the suboptimality criterion with feasibility guarantee will be derived chiefly.

Chapter 5

# $\epsilon$ Suboptimality Based Accelerated Termination for Linear Constrained Quadratic Optimization

In this chapter, to enlarge the application of the suboptimality criterion, the general linear constrained quadratic optimization problem is tackled. Equipped with cone programming (CP), the suboptimality criterion developed in this section can work without information of exact gradient or the optimal active set, enabling the methodology to be adopted in a broader scope. To highlight, the lower bound of suboptimality, which can be computed using only the initialization parameters, is demonstrated to be sufficient to make the iterative process terminated within finite iterations.

## 5.1 Introduction

Quadratic programming has long tracked massive interest in the society of control system, applied mathematics, and computer science, for it encompasses a large variety of applications, such as computational geometry, finance, process networks, robotics, telecommunications, energy, and data confidentiality, etc[21].

In general, iterative method is a main approach to solve QP due to its scalability and convergence feature. However, during the iterative process, the optimality is only guaranteed in the limit of iterations, and the feasibility is not ensured at any iterate. As a compromise, a suboptimal (with suboptimality to be determined depending on computation difficulty and time sensitivity) but feasible (in some applications for concerns of security or physical limits) solution is sometimes preferred over optimal solution in practical application. In [4], suboptimality focused on variables has been investigated, but

no definite criterion of objective value suboptimality has been proposed, and the method may generally fail under large suboptimality bound. In [28], though arbitrary suboptimality is fulfilled, solving an exponentially increasing number of QP is required to build the stopping criterion.

In this chapter, consolidating the iterative method and KKT criterion, a proactive method is proposed to solve for the optimal solution by dynamically identifying if the active set is optimal or not. Furthermore, the gradient norm based suboptimality in Chapter 2 is extended to general linear constrained quadratic optimization. This extension can deliver an arbitrary suboptimal and feasible solution, which requires no information of the optimal active set.

By transforming the original problem into only equality constrained problem using active set identified, whose explicit optimum and the best dual function value found during the iteration can be used to build $\epsilon$ suboptimality criterion. In the continuation, a cone programming (CP) incorporating feasibility check and $\epsilon$ suboptimality criterion can be used to produce a feasible $\epsilon$ suboptimal solution of the original problem. It must be highlighted that the identified active set only needs to be close enough, not necessarily identical, to the optimal active set to complete such a process, which enables the iterations to be terminated earlier than finding the optimal active set.

The main contributions of this chapter are threefold. First, the proactive method can deliver the optimal solution using a relative small scale of iterations compared to the traditional iterative method. Second, the suboptimal method can generate feasible solution of predefined suboptimality without knowing the optimal active set, which enables an even earlier termination of the iterative process prior to finding the optimal active set. Third, the lower bound of suboptimality has been demonstrated with finite iterations termination guarantee.

This chapter is organized as follows. §5.2 sets up the optimization problem and fundamentals. §5.3 proposes the proactive method combining the KKT criterion and general iterative approach. §5.4 illustrates transformations from active inequality constrains into equality ones, and proposes criterion to generate $\epsilon$ primal solution. Numerical experiments and results discussions are presented in §5.5. And conclusions are given in §5.6.

## 5.2 Problem statement and fundamentals

In this section, the linear constrained quadratic problem is formulated. Besides, $\epsilon$ primal solution, iteration mechanism, and definition of active set are introduced as fundamentals for methods studied later.

### 5.2.1 Problem statement and preliminaries

In this chapter, a general linear equality and inequality constrained quadratic problem is considered as below:

$$p^* = \min_{\boldsymbol{y}} \ \mathcal{J}(\boldsymbol{y}), \tag{5.1a}$$

$$\tilde{\boldsymbol{A}}\boldsymbol{y} = \tilde{\boldsymbol{b}}, \tag{5.1b}$$

$$\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{d}, \tag{5.1c}$$

where $\tilde{\boldsymbol{A}} \in \mathbb{R}^{n_e \times ny}$, $\text{rank}(\tilde{\boldsymbol{A}}) = n_r$, $n_r \in \mathbb{N}_{\geq 0}$, $\tilde{\boldsymbol{b}} \in \mathbb{R}^{n_e}$, $\boldsymbol{C} \in \mathbb{R}^{n_{ie} \times ny}$ and $\boldsymbol{d} \in \mathbb{R}^{n_{ie}}$, $\mathcal{J}(\boldsymbol{y}) = \frac{1}{2}||\boldsymbol{y}||^2_{\boldsymbol{R}}$, in this chapter $\boldsymbol{R}$ is assumed to be positive definite [1].

**Definition 5.** *The feasible set of problem* (5.1) *is defined as* $Y$*:*

$$Y = \{\boldsymbol{y} \in \mathbb{R}^{ny} \mid \tilde{\boldsymbol{A}}\boldsymbol{y} = \tilde{\boldsymbol{b}}, \boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{d}\}. \tag{5.2}$$

**Assumption 5.** *It is assumed that* $Y$ *is compact, closed, and not empty, and there is at least one* $\boldsymbol{y}$ *in the interior of* $Y$*. It is also assumed that* $n_r < ny$*.*

**Definition 6.** $\boldsymbol{y}$ *is said to be an* $\epsilon$ *primal solution of problem* (5.1) *if and only if* $\boldsymbol{y} \in Y$ *and*

$$\mathcal{J}(\boldsymbol{y}) - p^* \leq \epsilon. \tag{5.3}$$

The main objective of this chapter is to find an efficient way to solve an $\epsilon$ primal solution of problem (5.1).

1. Refer to [60] for case $\boldsymbol{R} \in \mathbb{S}^{ny}_{+}$, in which the technique does not change the method presented in this chapter.

### 5.2.2 Dual problem and iterative process

In this subsection, dual variables are introduced to form the dual function and dual problem associated with (5.1), which can be solved by a general iterative process.

To begin with, the dual problem and Lagrangian of problem (5.1) are defined as:

$$\tilde{g}^* = \max_{\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}\geq 0} \tilde{g}(\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}) = \max_{\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}\geq 0} \min_{\boldsymbol{y}} \tilde{\mathcal{L}}(\boldsymbol{y},\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}), \tag{5.4}$$
$$\tilde{\mathcal{L}}(\boldsymbol{y},\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}) = \frac{1}{2}||\boldsymbol{y}||^2_{\boldsymbol{R}} + \tilde{\boldsymbol{\theta}}^T(\tilde{\boldsymbol{A}}\boldsymbol{y} - \tilde{\boldsymbol{b}}) + \tilde{\boldsymbol{\lambda}}^T(\boldsymbol{C}\boldsymbol{y} - \boldsymbol{d}),$$

where $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{n_e}$ and $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}^{n_{ie}}_{\geq 0}$ are the dual variables associated with constraint (5.1b) and (5.1c) respectively.

**Remark 6.** *Since* $\boldsymbol{R} \in \mathbb{S}^{ny}_+$, (5.1b) *and* (5.1c) *are linear, problem* (5.1) *is convex. As the Slater's condition is satisfied by Assumption 5, the strong duality holds by Slater's theorem[8], namely* $p^* = \tilde{g}^*$.

A general gradient method is initiated to solve dual problem (5.4) as:

$$\tilde{\boldsymbol{\theta}}^{k+1} = \tilde{\boldsymbol{\theta}}^k + \alpha^k_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{A}}\boldsymbol{y}^k - \tilde{\boldsymbol{b}}), \tag{5.5a}$$
$$\tilde{\boldsymbol{\lambda}}^{k+1} = \max\{0, \tilde{\boldsymbol{\lambda}}^k + \alpha^k_{\tilde{\boldsymbol{\lambda}}}(\mathbf{C}\boldsymbol{y}^k - \mathbf{d})\}, \tag{5.5b}$$
$$\boldsymbol{y}^{k+1} = -\boldsymbol{R}^{-1}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\theta}}^{k+1} + \boldsymbol{C}^T\tilde{\boldsymbol{\lambda}}^{k+1}), \tag{5.5c}$$

where $\alpha^k_{\tilde{\boldsymbol{\theta}}}, \alpha^k_{\tilde{\boldsymbol{\lambda}}} \in \mathbb{R}_{>0}$ are step size associated with $\tilde{\boldsymbol{\theta}}^k$ and $\tilde{\boldsymbol{\lambda}}^k$ respectively, and (5.5c) is obtained by subsituting $\tilde{\boldsymbol{\lambda}}^{k+1}$ and $\tilde{\boldsymbol{\theta}}^{k+1}$ into $\nabla_{\boldsymbol{y}}\tilde{\mathcal{L}}(\boldsymbol{y},\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\lambda}}) = \mathbf{0}$.

**Assumption 6.** *It is assumed that* $\alpha^k_{\tilde{\boldsymbol{\theta}}}$ *and* $\alpha^k_{\tilde{\boldsymbol{\lambda}}}$ *satisfy one of step size conditions: minimization rule, Armijo rule and diminishing stepsize [6], such that the sequence* $\{\boldsymbol{y}^k\}$ *converges to the optimal solution of problem* (5.1)*, namely*

$$\lim_{k\to\infty} \boldsymbol{y}^k = \boldsymbol{y}^*. \tag{5.6}$$

Note that iterative process (5.5) alone cannot generate $\epsilon$ primal solution of problem (5.1), since $\boldsymbol{y}^k \in Y$ is not guaranteed at any iterate.

Let $\mathcal{P} = \{1, ..., n_{ie}\}$, definitions of active constraint, active set and inactive set are given as follows.

**Definition 7.** *During iterative process* (5.5)*, the i-th constraint of* (5.1c)*:* $\boldsymbol{C}_i\boldsymbol{y} \leq \boldsymbol{d}_i$ *(*$\boldsymbol{C} = \bigoplus_{i=1}^{n_{ie}} \boldsymbol{C}_i$*,* $\boldsymbol{d} = \bigoplus_{i=1}^{n_{ie}} \boldsymbol{d}_i$*), is said to be active at* $\boldsymbol{y}^k$ *if* $\boldsymbol{C}_i\boldsymbol{y}^k \geq \boldsymbol{d}_i$*, i.e., the equality is reached or the inequality is violated; or* $\tilde{\boldsymbol{\lambda}}_i^k > 0$*, its corresponding dual variable is turned positive. Denote* $\mathcal{A}^k$ *and* $\mathcal{I}^k$ *the active and inactive set of constraints* (5.1c) *at k-th iteration, which are defined as:*

$$\mathcal{A}^k = \{i \in \mathcal{P} \mid \boldsymbol{C}_i\boldsymbol{y}^k \geq \boldsymbol{d}_i \text{ or } \tilde{\boldsymbol{\lambda}}_i^k > 0\}, \tag{5.7}$$

$$\mathcal{I}^k = \mathcal{P}\backslash\mathcal{A}^k. \tag{5.8}$$

Let $\boldsymbol{y}^*$ denote the optimal solution of problem (5.1), $\mathcal{A}^*$ and $\mathcal{I}^*$ are used to denote the optimal active and inactive set, $\mathcal{A}^* = \{i \in \mathcal{P} \mid \boldsymbol{C}_i\boldsymbol{y}^* = \boldsymbol{d}_i\}$ [2], $\mathcal{I}^* = \mathcal{P}\backslash\mathcal{A}^*$.

### 5.2.3 KKT criterion for active set

Below, a linear programming (LP) based on KKT condition [26] can be used to test if $\mathcal{A}^k$ is the optimal active set of problem (5.1), whose solution is denoted as $(\boldsymbol{y}^*, \tilde{\boldsymbol{\lambda}}^*, \boldsymbol{s}^*, h^*)$ if it exists.

$$\min_{\boldsymbol{y},\tilde{\boldsymbol{\lambda}}_{\mathcal{A}^k},\boldsymbol{s}_{\mathcal{I}^k},h} \quad -h \tag{5.9a}$$

$$s.t.\ \boldsymbol{R}\boldsymbol{y} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\theta}} + \boldsymbol{C}_{\mathcal{A}^k}^T\tilde{\boldsymbol{\lambda}}_{\mathcal{A}^k} = \boldsymbol{0}, \tag{5.9b}$$

$$\tilde{\boldsymbol{A}}\boldsymbol{y} - \tilde{\boldsymbol{b}} = \boldsymbol{0}, \tag{5.9c}$$

$$\boldsymbol{C}_{\mathcal{A}^k}\boldsymbol{y} - \boldsymbol{d}_{\mathcal{A}^k} = \boldsymbol{0}, \tag{5.9d}$$

$$\boldsymbol{C}_{\mathcal{I}^k}\boldsymbol{y} - \boldsymbol{d}_{\mathcal{I}^k} + \boldsymbol{s}_{\mathcal{I}^k} = \boldsymbol{0}, \tag{5.9e}$$

$$he_1 \leq \tilde{\boldsymbol{\lambda}}_{\mathcal{A}^k}, \tag{5.9f}$$

$$he_2 \leq \boldsymbol{s}_{\mathcal{I}^k}, \tag{5.9g}$$

$$0 \leq h, \tag{5.9h}$$

where $\boldsymbol{C}_{\mathcal{A}^k} = \bigoplus_{i\in\mathcal{A}^k} \boldsymbol{C}_i$, $\boldsymbol{C}_{\mathcal{A}^k} \in \mathbb{R}^{c_k \times ny}$, $\boldsymbol{d}_{\mathcal{A}^k} = \bigoplus_{i\in\mathcal{A}^k} \boldsymbol{d}_i$, $\boldsymbol{d}_{\mathcal{A}^k} \in \mathbb{R}^{c_k}$, $\boldsymbol{C}_{\mathcal{I}^k} = \bigoplus_{i\in\mathcal{I}^k} \boldsymbol{C}_i$, $\boldsymbol{d}_{\mathcal{I}^k} = \bigoplus_{i\in\mathcal{I}^k} \boldsymbol{d}_i$, $e_1$ and $e_2$ are column vectors of ones of appropriate size.

It is not assumed that the strict complementarity condition or the linear independence constraint qualification (LICQ) is satisfied [3] in this chapter, where the former is said to

2. It is felicitous to replace "$\geq$" with "=", and remove the dual variable check at the optimal solution.
3. The notion of LICQ and strict complementarity condition are borrowed from [43].

hold if matrix $\tilde{\boldsymbol{A}}_{\mathcal{A}^k}$ ($\tilde{\boldsymbol{A}}_{\mathcal{A}^k} = \tilde{\boldsymbol{A}} \oplus \boldsymbol{C}_{\mathcal{A}^k}$) has full row rank and the latter is said to hold if $\lambda_i > 0$ for each $\boldsymbol{C}_i\boldsymbol{y}^* = \boldsymbol{d}_i$. As a result, $\mathcal{A}^*$ is not necessary to be unique. Nonetheless, LP (5.9) is still valid to generate $\boldsymbol{y}^*$.

## 5.3 Proactive optimal active set identification method (POASIM)

In this section, combining iterative process and KKT criterion, a method is proposed to solve problem (5.1) by proactively checking active set generated in iterative process, which fundamentally requires fewer iterations compared to the conventional iterative method (result of a illustrative example is presented in §5.5.1).

This proactive algorithm is summarized in Alg. 8. Note that $\mathcal{A}^k$ and $\mathcal{A}^{k-i}$ ($i = 1, ...k$) are possible to be identical, it is sufficient to only test $\mathcal{A}^k$ that has not been tested before. The optimal solution of problem (5.5.1) can be obtained if the optimal active set is identified, which, however, cannot be guaranteed to happen in implementing POASIM. Therefore, it is one crucial drawback of POASIM, and it can be overcome by the suboptimal method proposed in the next section.

**Algorithm 8** Proactive Optimal Active Set Identification Method (POASIM)

1: Initialize: $\boldsymbol{\theta}^{-1}$, $\boldsymbol{\lambda}^{-1}$, $k = 0$ and $\epsilon$. $\boldsymbol{y}^{-1}$ is obtained by (5.5c).
2: **repeat**
3: Update primal and dual variables by (5.5), update $\mathcal{A}^k$ by (5.7)
4: **if** LP (5.9) has a solution **then return** $\boldsymbol{y}^*$
5: **end if**
6: $k \leftarrow k + 1$
7: **until** $\boldsymbol{y}^*$ is returned

## 5.4 Active set based $\epsilon$ suboptimal approach

In this section, based on Definition 7, problem (5.1) is dynamically converted into equality constrained formulation during the iteration, whose optimal solution can be solved explicitly. Together with the best dual objective value of problem (5.4), it can be checked that whether an $\epsilon$ primal solution of problem (5.1) is achievable under $\mathcal{A}^k$ .

### 5.4.1 Transforming active constraints into equality constraints

Definition 7 can be used to identify the active and inactive constraints of (5.1c) at $k$-th iteration. By doing so, Problem (5.1) can be dynamically converted into the equality constrained problem (denote its optimizer as $\boldsymbol{y}^*_{\mathcal{A}^k}$) below:

$$\mathcal{J}^*_{\mathcal{A}^k} = \min_{\boldsymbol{y}} \ \mathcal{J}(\boldsymbol{y}) \tag{5.10a}$$

$$s.t. \ \tilde{\boldsymbol{A}}_{\mathcal{A}^k}\boldsymbol{y} = \tilde{\boldsymbol{b}}_{\mathcal{A}^k}, \tag{5.10b}$$

where $\tilde{\boldsymbol{A}}_{\mathcal{A}^k} = \tilde{\boldsymbol{A}} \oplus \boldsymbol{C}_{\mathcal{A}^k}$, and $\tilde{\boldsymbol{b}}_{\mathcal{A}^k} = \tilde{\boldsymbol{b}} \oplus \boldsymbol{d}_{\mathcal{A}^k}$.

**Lemma 5.** *$\boldsymbol{y}^*_{\mathcal{A}^k}$ can be solved by the linear equation group below:*

$$\begin{cases} \tilde{\boldsymbol{A}}_{\mathcal{A}^k}\boldsymbol{y} = \tilde{\boldsymbol{b}}_{\mathcal{A}^k}, \\ \boldsymbol{F}^T_{\mathcal{A}^k}\boldsymbol{R}\boldsymbol{y} = \boldsymbol{0}, \end{cases} \tag{5.11}$$

*where $\boldsymbol{F}_{\mathcal{A}^k} \in \mathbb{R}^{ny\times(ny-n_r-c_k)}$ is a orthonormal null space matrix of $\tilde{\boldsymbol{A}}_{\mathcal{A}^k}$ satisfying $\tilde{\boldsymbol{A}}_{\mathcal{A}^k}\boldsymbol{F}_{\mathcal{A}^k} = \boldsymbol{0}$.*

*Proof.* By Assumption 5, $\text{rank}(\boldsymbol{F}^T_{\mathcal{A}^k}) = ny - n_r - c_k$. Since $\text{rank}(\tilde{\boldsymbol{A}}_{\mathcal{A}^k}) = c_k + n_r$ and $\boldsymbol{R} \in \mathbb{S}^{ny}_{++}$, the linear equation group (5.11) has row rank as $ny$, which means it has the unique solution.

Here, the feasible set $Y_{\mathcal{A}^k}$ of problem (5.10) is characterized as:

$$\begin{aligned} Y_{\mathcal{A}^k} =& \{\boldsymbol{y} \in \mathbb{R}^{ny} \mid \tilde{\boldsymbol{A}}_{\mathcal{A}^k}\boldsymbol{y} = \tilde{\boldsymbol{b}}_{\mathcal{A}^k}\} \\ =& \{\hat{\boldsymbol{y}}_{\mathcal{A}^k} + \boldsymbol{F}_{\mathcal{A}^k}\boldsymbol{t}_{\mathcal{A}^k} \mid \boldsymbol{t}_{\mathcal{A}^k} \in \mathbb{R}^{ny-n_r-c_k}\}, \end{aligned} \tag{5.12}$$

where the second equation is based on any $\hat{\boldsymbol{y}}_{\mathcal{A}^k} \in Y_{\mathcal{A}^k}$.

It is trivial that the solution of the linear equation group is a feasible solution of problem (5.10) since (5.10b) is satisfied.

Next, the optimality will be proved. By (5.12), problem (5.10) is equivalent to:

$$\boldsymbol{J}^*_{\mathcal{A}^k} = \min_{\boldsymbol{t}_{\mathcal{A}^k}} \boldsymbol{J}(\boldsymbol{t}_{\mathcal{A}^k}), \tag{5.13}$$

where $\boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k}) = \frac{1}{2}||\hat{\boldsymbol{y}}_{\mathcal{A}^k} + \boldsymbol{F}_{\mathcal{A}^k}\boldsymbol{t}_{\mathcal{A}^k}||^2_{\boldsymbol{R}}$, and $\boldsymbol{J}^*_{\mathcal{A}^k} = \mathcal{J}^*_{\mathcal{A}^k}$.

Accordingly, it gives $\nabla\boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k}) = \boldsymbol{F}^T_{\mathcal{A}^k}\boldsymbol{R}(\hat{\boldsymbol{y}}_{\mathcal{A}^k} + \boldsymbol{F}_{\mathcal{A}^k}\boldsymbol{t}_{\mathcal{A}^k})$, $\nabla^2\boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k}) = \boldsymbol{F}^T_{\mathcal{A}^k}\boldsymbol{R}\boldsymbol{F}_{\mathcal{A}^k}$. For $\forall \boldsymbol{t}_{\mathcal{A}^k} \in \mathbb{R}^{ny-n_r-c_k}$, a feasible solution of (5.10b) can be assigned, denoted as $\boldsymbol{y}_{\mathcal{A}^k}$, to

have $\boldsymbol{y}_{\mathcal{A}^k} = \hat{\boldsymbol{y}}_{\mathcal{A}^k} + \boldsymbol{F}_{\mathcal{A}^k}\boldsymbol{t}_{\mathcal{A}^k}$, then it gives $\nabla \boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k}) = \boldsymbol{F}_{\mathcal{A}^k}^T \boldsymbol{R}\boldsymbol{y}_{\mathcal{A}^k}$.

At last, the necessary and sufficient optimality condition of unconstrained convex optimization (5.13) is given as: $||\nabla \boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k})|| = 0$. And the proof can be concluded by (5.11). □

### 5.4.2 Active set based $\epsilon$ suboptimal criterion

To emphasis the main contribution, henceforth the methodology only for case $c_k + n_r < ny$ will be delineated [4]. Let $\tilde{g}^k_{best} = \sup_{i \le k, i \in \mathbb{N}_{>0}} \tilde{g}(\tilde{\boldsymbol{\theta}}^i, \tilde{\boldsymbol{\lambda}}^i)$. Since the solution of problem (5.10) is not unique, the relation between $\tilde{g}^k_{best}$ and $\boldsymbol{y}^*_{\mathcal{A}^k}$ is needed to build the $\epsilon$ suboptimality criterion. By the primal-dual theory[8], it gives $\tilde{g}^k_{best} \le p^*$. Combining with Definition 6, if $\boldsymbol{y} \in Y$, and satisfies

$$\mathcal{J}(\boldsymbol{y}) - \tilde{g}^k_{best} \le \epsilon, \tag{5.14}$$

then $\boldsymbol{y}$ is an $\epsilon$ primal solution of problem (5.5.1).

In solving an $\epsilon$ primal solution of problem (5.1), condition (5.14) can further reduce the gap between $\tilde{g}^k_{best}$ and $p^*$ by taking advantage of the sequence $\{\tilde{g}^k_{best}\}$ generated. Since the solution of problem (5.10) is not unique, we need the relation between $\tilde{g}^k_{best}$ and $\boldsymbol{y}^*_{\mathcal{A}^k}$ as a base to build the criterion for $\epsilon$ suboptimality.

There are 3 possibilities between $\mathcal{J}^*_{\mathcal{A}^k}$ and $\tilde{g}^k_{best}$ depending on the identification correctness of $\mathcal{A}^k$ and the gap between $\tilde{g}^k_{best}$ and $\mathcal{J}^*$:

$$\mathcal{J}^*_{\mathcal{A}^k} < \tilde{g}^k_{best}, \tag{5.15}$$

$$0 \le \mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best} \le \epsilon, \tag{5.16}$$

$$\epsilon < \mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best}. \tag{5.17}$$

Note that in case (5.17), no definite $\epsilon$ suboptimality criterion can be devised without knowing $\mathcal{A}^*$, since even $\mathcal{J}^*_{\mathcal{A}^k}$, the optimum by far fails (5.14). The following 2 propositions will be used to build criteria for $\epsilon$ suboptimality when (5.15) or (5.16) is satisfied.

**Proposition 3.** *Given* $\Delta \in R_{>0}$, *if* $\boldsymbol{y}$ *satisfies* (5.10b) *and* $||\boldsymbol{F}_{\mathcal{A}^k}^T \boldsymbol{R}\boldsymbol{y}|| \le (2\beta_{\mathcal{A}^k}\Delta)^{\frac{1}{2}}$, *where*

4. The case $c_k + n_r > ny$ is unsolvable, executing $\mathcal{A}^k \leftarrow \mathcal{A}^k \backslash \{i \in \mathcal{A}^k \mid \tilde{\boldsymbol{\lambda}}^k_i < \tilde{\boldsymbol{\lambda}}'^k\}$ ($\tilde{\boldsymbol{\lambda}}'^k$ denote the $ny$-th largest value of $\tilde{\boldsymbol{\lambda}}^k_i$, $i \in \mathcal{A}^k$), it can be converted into case $c_k + n_r = ny$, in which case the only solution can be obtained by solving linear equation (5.10b).

$\beta_{\mathcal{A}^k} = \min \text{eig}(\boldsymbol{F}_{\mathcal{A}^k}^T \boldsymbol{R} \boldsymbol{F}_{\mathcal{A}^k})$, *then it must have:*

$$\mathcal{J}(\boldsymbol{y}) - \mathcal{J}_{\mathcal{A}^k}^* \leq \Delta. \tag{5.18}$$

*Proof.* As $\boldsymbol{y}$ satisfies (5.10b), it is a feasible solution of problem (5.10). Since problem (5.13) is convex and unconstrained, and $\nabla^2 \boldsymbol{J}_{\mathcal{A}^k}(\boldsymbol{t}_{\mathcal{A}^k})$ is lower bounded by $\beta_{\mathcal{A}^k}$ (because $\boldsymbol{A}_{\mathcal{A}^k}$ is full row rank), then (5.18) holds by applying (9.10) of [8]. □

Now, consider the CP below, and denote its optimizer as $(\boldsymbol{y}_{sub}, s_{\mathcal{I}^k}^1, h^1)$ if it exists.

$$\min_{\boldsymbol{y}, s_{\mathcal{I}^k}, h} \quad -h \tag{5.19a}$$

$$s.t.\ (5.10b), (5.9e), (5.9g), (5.9h),$$

$$||\boldsymbol{F}_{\mathcal{A}^k}^T \boldsymbol{R} \boldsymbol{y}|| \leq (2\beta_{\mathcal{A}^k}(\epsilon + \tilde{g}_{best}^k - \mathcal{J}_{\mathcal{A}^k}^*))^{\frac{1}{2}}. \tag{5.19b}$$

Based on $\epsilon$ suboptimality criteria demonstrated in Proposition 3, CP (5.19) can be used to obtain an $\epsilon$ primal solution of problem (5.1), which will be illustrated in the following proposition.

**Proposition 4.** *If LP* (5.19) *has a solution and*

$$\mathcal{J}_{\mathcal{A}^k}^* - \tilde{g}_{best}^k \leq \epsilon, \tag{5.20}$$

*then* $\boldsymbol{y}_{sub}$ *is an* $\epsilon$ *primal solution of problem* (5.1).

*Proof.* First, consider case (5.15), let $\delta^k = \tilde{g}_{best}^k - \mathcal{J}_{\mathcal{A}^k}^*$. As $\boldsymbol{y}_{sub}$ satisfies (5.10b),(5.9e), (5.9g), (5.9h), it gives $\boldsymbol{A}_{\mathcal{A}^k} \boldsymbol{y}_{sub} = \boldsymbol{b}_{\mathcal{A}^k}$, and $\boldsymbol{C}_{\mathcal{I}^k} \boldsymbol{y}_{sub} \leq \boldsymbol{d}_{\mathcal{I}^k}$. Since at each iteration, $\boldsymbol{C}$ consists of $\boldsymbol{C}_{\mathcal{I}^k}$ and $\boldsymbol{C}_{\mathcal{A}^k}$, then by (5.2), $\boldsymbol{y}_{sub} \in Y$. Subsequently, by the primal-dual theory[8], it gives

$$0 \leq \mathcal{J}(\boldsymbol{y}_{sub}) - \tilde{g}_{best}^k. \tag{5.21}$$

As (5.19b) is satisfied by $\boldsymbol{y}_{sub}$, it holds by Proposition 3 that:

$$0 \leq \mathcal{J}(\boldsymbol{y}_{sub}) - \tilde{g}_{best}^k \leq \epsilon. \tag{5.22}$$

As a consequence, combining $\boldsymbol{y}_{sub} \in Y$, (5.21) and (5.22), $\boldsymbol{y}_{sub}$ is an $\epsilon$ primal solution of problem (5.1) by (5.14).

Second, consider case (5.16), let $\Delta^k = \mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best}$, by (5.18) it holds that: $\mathcal{J}(\boldsymbol{y}) - \mathcal{J}^*_{\mathcal{A}^k} \leq \epsilon - \Delta^k$. Namely, $\mathcal{J}(\boldsymbol{y}) - \tilde{g}^k_{best} \leq \epsilon$. The rest of the proof remains the same with that of case (5.15). □

In a cost ascending order, Alg. 9 presents the complete algorithm to compute an $\epsilon$ primal solution of problem (5.1): first check whether $\boldsymbol{y}^*_{\mathcal{A}^k}$ is an $\epsilon$ primal solution; if not, then check whether an $\epsilon$ primal solution can be found by CP (5.19) using $\tilde{g}^k_{best}$.

**Algorithm 9** Active Set Based Suboptimal Algorithm (ASBSA)

Initialize: $\boldsymbol{\theta}^{-1}$, $\boldsymbol{\lambda}^{-1}$, $k = 0$ and $\epsilon$. $\boldsymbol{y}^{-1}$ is obtained by (5.5c)
**repeat**
  Update primal and dual variables by (5.5), update $\mathcal{A}^k$ by (5.7), compute $\boldsymbol{y}^*_{\mathcal{A}^k}$ by (5.11)
  **if** $\boldsymbol{y}^*_{\mathcal{A}^k} \in Y$ **then**
    **if** $\boldsymbol{y}^*_{\mathcal{A}^k}$ satisfies (5.14) **then return** $\boldsymbol{y}^*_{\mathcal{A}^k}$
    **end if**
  **end if**
  **if** $\boldsymbol{y}^*_{\mathcal{A}^k}$ satisfies (5.20) **then**
    **if** CP (5.19) has a solution **then return** $\boldsymbol{y}_{sub}$
    **end if**
  **end if**
  $k \leftarrow k + 1$
**until** one of $\boldsymbol{y}^*_{\mathcal{A}^k}$ and $\boldsymbol{y}_{sub}$ is returned

### 5.4.3 Optimization properties of ASBSA

From here, the following 2 lemmas will be used to derive the lower bound of $\epsilon$: for any value above the bound, ASBSA can terminate within finite iterations.

**Lemma 6.** *Under Assumption 6, in implementing ASBSA,* (5.20) *can be satisfied within finite iterations for* $\forall \epsilon > 0$.

*Proof.* First, consider the following problem: $\delta = \inf_{i \in \mathcal{I}^*} \{\min_{\boldsymbol{y}} ||\boldsymbol{y} - \boldsymbol{y}^*|| \mid \boldsymbol{C}_i \boldsymbol{y} = \boldsymbol{d}_i\}$. Then, as $\boldsymbol{y}^k$ asymptotically converges to $\boldsymbol{y}^*$ by Assumption 6, there exists a $k_1$ such that $\forall k \geq k_1$, $||\boldsymbol{y}^k - \boldsymbol{y}^*|| \leq \delta$. So, for $\mathcal{A}^k$ generated by (5.7), $\forall k \geq k_1$, it gives $\mathcal{I}^* \cap \mathcal{A}^k = \emptyset$. Therefore, it gives $\mathcal{A}^k \subset \mathcal{A}^*$ for $\forall k \geq k_1$, which means $Y$ is a proper subset of the feasible set of problem (5.10), thus it must holds that:

$$\mathcal{J}^*_{\mathcal{A}^k} \leq \mathcal{J}^*, \ \forall k \geq k_1. \tag{5.23}$$

For a given $\epsilon > 0$, there exists a $k_2$ by Remark 6 such that

$$\mathcal{J}^* - \tilde{g}^k_{best} \leq \epsilon, \ \forall k \geq k_2. \tag{5.24}$$

Consequently, combing (5.23) and (5.24), it gives

$$\mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best} \leq \epsilon, \ \forall k \geq \max\{k_1, k_2\}. \tag{5.25}$$

And this completes the proof. □

Here, consider the following norm minimization:

$$\bar{\delta} = \min_{\boldsymbol{y}} \ ||\boldsymbol{F}_{\boldsymbol{A}}^T \boldsymbol{R}\boldsymbol{y}||, \ s.t. \ \boldsymbol{y} \in Y,$$

where $\boldsymbol{F}_{\boldsymbol{A}} \in \mathbb{R}^{ny \times (ny - n_r)}$ is a orthonormal null space matrix of $\boldsymbol{A}$ satisfying $\boldsymbol{A}\boldsymbol{F}_{\boldsymbol{A}} = \boldsymbol{0}$.

**Lemma 7.** *Under Assumption 6, for $\forall \epsilon > \bar{\delta}/2\beta_{\boldsymbol{A}}$, where $\beta_{\boldsymbol{A}} = \min \text{eig}(\boldsymbol{F}_{\boldsymbol{A}}^T \boldsymbol{R} \boldsymbol{F}_{\boldsymbol{A}})$, an $\epsilon$ suboptimal solution of problem* (5.1) *can be generated within finite iterations in implementing ASBSA.*

*Proof.* By Lemma 6, given an arbitrary $\epsilon' > 0$, there exist a $k$ such that

$$\mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best} \leq \epsilon'. \tag{5.26}$$

Next, consider the problem:

$$\delta^k = \min_{\boldsymbol{y}} \ ||\boldsymbol{F}_{\mathcal{A}^k}^T \boldsymbol{R}\boldsymbol{y}||, \ s.t. \ (5.10b), \boldsymbol{C}_{\mathcal{I}^k}\boldsymbol{y} \leq \boldsymbol{d}_{\mathcal{I}^k}.$$

Observing CP (5.19), for $\epsilon^k = (\delta^k)^2/2\beta_{\mathcal{A}^k} + \mathcal{J}^*_{\mathcal{A}^k} - \tilde{g}^k_{best}$, an $\epsilon^k$ suboptimal solution of problem (5.1) can be found by solving CP (5.19).

By (5.26), an $((\delta^k)^2/2\beta_{\mathcal{A}^k} + \epsilon')$ suboptimal solution of problem (5.1) can be generated with at most $k$ iterations. If it can be shown that

$$(\delta^k)^2/2\beta_{\mathcal{A}^k} \leq \bar{\delta}^2/2\beta_{\boldsymbol{A}}, \tag{5.27}$$

then for $\epsilon = \bar{\delta}/2\beta_{\boldsymbol{A}} + \epsilon'$, an $\epsilon$ suboptimal solution of problem (5.1) can be generated with at most $k$ iterations. Since $\epsilon'$ can be arbitrary small, we have that for $\forall \epsilon > \bar{\delta}/2\beta_{\boldsymbol{A}}$, an $\epsilon$ suboptimal solution of problem (5.1) can be generated within finite iteration.

The proof of (5.27) is given from here. If $\boldsymbol{A}_{\mathcal{A}^k} = \boldsymbol{A}$, (5.27) trivially holds. Considering the case $\boldsymbol{A}_{\mathcal{A}^k} \neq \boldsymbol{A}$, it indicates that $\boldsymbol{A}_{\mathcal{A}^k} = \boldsymbol{A} \oplus \boldsymbol{C}_{\mathcal{A}^k}$. Subsequently, by $\boldsymbol{AF} = \boldsymbol{0}$, $\boldsymbol{AF}_{\mathcal{A}^k} = \boldsymbol{0}$, and $\boldsymbol{C}_{\mathcal{A}^k}\boldsymbol{F}_{\mathcal{A}^k} = \boldsymbol{0}$, the null space of $\boldsymbol{A}_{\mathcal{A}^k}$ is a subspace of the null space of $\boldsymbol{A}$. Since $\boldsymbol{F} \in \mathbb{R}^{ny\times(ny-n_r)}$ and $\boldsymbol{F}_{\mathcal{A}^k} \in \mathbb{R}^{ny\times(ny-n_r-c_k)}$, there exists a semi-orthogonal matrix $P \in \mathbb{R}^{(ny-n_r)\times(ny-n_r-c_k)}$ with $P^TP = \boldsymbol{I}$, such that $\boldsymbol{F}P = \boldsymbol{F}_{\mathcal{A}^k}$. It follows $\boldsymbol{F}_{\mathcal{A}^k}^T\boldsymbol{R}\boldsymbol{F}_{\mathcal{A}^k} = P^T\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{F}P$. Then by Poincaré separation theorem[32],

$$\beta_{\boldsymbol{A}} \leq \beta_{\mathcal{A}^k}. \tag{5.28}$$

For any $\boldsymbol{y} \in Y$,

$$||\boldsymbol{F}_{\mathcal{A}^k}^T\boldsymbol{R}\boldsymbol{y}|| = ||P^T\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{y}|| \leq ||P^T||||\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{y}|| = ||\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{y}||, \tag{5.29}$$

where the inequality uses Cauchy-Shwarz inequality, and the equality uses the property of semi-orthogonal matrix that $||P^T|| = ||P|| = 1$.

Finally, (5.27) can be concluded by (5.28) and (5.29), and this completes the proof. □

**Remark 7.** *Note that $\epsilon > \bar{\delta}/2\beta_{\boldsymbol{A}}$ is a sufficient condition for ASBSA to be terminated within finite iterations. In practice, it is possible to take $\epsilon$ much lower than $\bar{\delta}/2\beta_{\boldsymbol{A}}$, which will be illustrated with a numerical example in §5.5.1.*

Regarding implementation of ASBSA, economic computation techniques as follows can further improve its efficiency:

1. for each distinct $\mathcal{A}^k$, its according variables $\boldsymbol{C}_{\mathcal{A}^k}, \boldsymbol{d}_{\mathcal{A}^k}, \boldsymbol{C}_{\mathcal{I}^k}, \boldsymbol{d}_{\mathcal{I}^k}, \boldsymbol{y}^*_{\mathcal{A}^k}, \boldsymbol{F}_{\mathcal{A}^k}$ and $\mathcal{J}^*_{\mathcal{A}^k}$ can be stored, then if $\mathcal{A}^{k+i} = \mathcal{A}^k, i = 1, 2, ...$, the above mentioned variables can be retrieved from the stored data, instead to compute from the scratch [5];
2. it requires that $\mathcal{A}^k$ in Step 9 of ASBSA has not been tested by CP (5.19) before;
3. at each iteration, if $\boldsymbol{y}^*_{\mathcal{A}^k}$ satisfies (5.14) and $\boldsymbol{y}^*_{\mathcal{A}^k} \in Y$, then ASBSA can be terminated without excess computation. In addition, to avoid unnecessary solving of CP (5.19), for each $\boldsymbol{y}^*_{\mathcal{A}^k} \in Y$ and $\mathcal{J}(\boldsymbol{y}) - g^{\bar{k}}_{best} > \epsilon$, denote $D_{\mathcal{A}^k} = \mathcal{J}^*_{\mathcal{A}^k} - \epsilon$, then at every $\bar{k} > k$, we can claim that $\boldsymbol{y}^*_{\mathcal{A}^k}$ is an $\epsilon$ suboptimal solution of problem (5.1) if $g^{\bar{k}}_{best} \geq D_{\mathcal{A}^k}$.

5. This technique can also be applied to POASIM.

## 5.5 Numerical experiments

In the numerical experiments, Nesterov gradient descent[42] [24] is adopted as follows for iteration (5.5a) (5.5b), which is proved to be the best first order gradient method [41].

$$\tilde{\boldsymbol{\theta}}^{k+1} = \hat{\boldsymbol{\theta}}^k + \frac{1}{L}(\tilde{\boldsymbol{A}}\hat{\boldsymbol{y}}^k - \tilde{\boldsymbol{b}}),$$
$$\tilde{\boldsymbol{\lambda}}^{k+1} = \max\{0, \hat{\boldsymbol{\lambda}}^k + \frac{1}{L}(\mathbf{C}\hat{\boldsymbol{y}}^k - \mathbf{d})\},$$

where for a vector $\boldsymbol{\nu}$, $\hat{\boldsymbol{\nu}}^k = \boldsymbol{\nu}^k + \frac{k-1}{k+2}(\boldsymbol{\nu}^k - \boldsymbol{\nu}^{k-1})$, $L = ||\boldsymbol{E}\boldsymbol{R}^{-1}\boldsymbol{E}^T||_2$, and $\boldsymbol{E} = (\tilde{\boldsymbol{A}}^T, \boldsymbol{C}^T)^T$.

Specifically, 2 groups of experiments are carried out: 1. single small size problem for a clear-cut comparison of time and iteration number magnitude among 2 algorithms proposed (POASIM and ASBSA) and pure iterative process; 2. for each $ny$ equals $10, 20, 50, 100,$ and 200, 1000 randomly generated tests for general performance comparison between POASIM and ASBSA. All numerical experiments are carried out using MATLAB 2020b on a Windows 10 PC with 2.20 GHz Core i7-8750H CPU and 16GB RAM.

In detail, $0 \leq \boldsymbol{y} \leq 1$ is set for inequality constraints (5.1c), $n_e = ny/2$, the sparsity concerning matrix $\tilde{\boldsymbol{A}}$ is randomly drawn from uniform distribution $(0, 1)$ of each problem, and each non zero entry of $\tilde{\boldsymbol{A}}$ is randomly drawn from uniform distribution $(-0.5, 0.5)$, and the i-th element of $\tilde{\boldsymbol{b}}$ is randomly drawn from uniform distribution $(0, \tilde{\boldsymbol{A}}_i \cdot \mathbf{1})$ to make problem (5.1) feasible, where $\tilde{\boldsymbol{A}}_i$ and $\mathbf{1}$ denote the $i$-th row of $\tilde{\boldsymbol{A}}$ and column vector of ones of appropriate size respectively. The penalty matrix is set as $\boldsymbol{R} = \boldsymbol{I}$.

### 5.5.1 Single test for comparison among 4 methods

To initiate a perception of 2 newly proposed algorithms (POASIM, ASBSA) and pure iterative process (5.5), here the comparison is presented by small size (ny=10) test, whose parameter setting can be found in Appendix B. From Table. 5.1, when $\epsilon$ suboptimality is only concerned, Nesterov gradient descent discloses evident superiority in iteration number and time. However, if feasibility is required, Nesterov gradient descent requires 5000 times more iterations, which results in worse time performance than 2 proposed methods. POASIM and ASBSA, promised to deliver optimal (of course feasible) solutions, reveal favorable results in iteration number and time, and hence will be investigated further with randomly generated problems of larger size in the following subsection.

Note that, in this test, $\bar{\delta}/2\beta_{\boldsymbol{A}} = 0.0348$, if take $\epsilon = \bar{\delta}/2\beta_{\boldsymbol{A}}$, then the corresponding

Table 5.1 – Performance comparison among Nesterov gradient descent, POASIM, and ASBSA of a single test
with ny=10, and predefined relative $\epsilon$ as $1 \times 10^{-2}$

| | Nesterov | | POASIM | ASBSA |
|---|---|---|---|---|
| | $\epsilon$* | $\epsilon + Y$** | | |
| Computation time (s) | 5.8 $\times 10^{-3}$ | 4.3 $\times 10^{-1}$ | 8.9 $\times 10^{-2}$ | 1.7 $\times 10^{-2}$ |
| # of iterations or (5.9) solved | 15 | 74803 | 20 | 15 |
| Primal feasible | no | yes | yes | yes |

* solved by (5.5a)-(5.5b), a posterior $\epsilon$ suboptimality criterion is used: $p^* - \tilde{g}(\tilde{\boldsymbol{\theta}}^k, \tilde{\boldsymbol{\lambda}}^k) \leq 0.01p^*$, where $p^*$ is known as a parameter.
** solved by (5.5a)-(5.5b), $\epsilon$ suboptimality criterion: $p^* - \tilde{g}(\tilde{\boldsymbol{\theta}}^k, \tilde{\boldsymbol{\lambda}}^k) \leq 0.01p^*$, feasibility criterion: $||\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}|| \leq 1 \times 10^{-16}$, $\boldsymbol{C}\boldsymbol{y} - \boldsymbol{d} \leq 1 \times 10^{-16}$, the magnitude $1 \times 10^{-16}$ is computed from optimal solution $\boldsymbol{y}^*$ that solved by POASIM.

Table 5.2 – Random tests statistics of ASBSA: average and maximal relative error
The magnitude of $1\times10^{-4}$, $1\times10^{-3}$, $1\times10^{-2}$ and $1\times10^{-1}$ are omitted from the results according to the relative suboptimality referred for space-saving. As a consequence, any result presented with value less than 1 means that predefined suboptimality is fulfilled.

| | Ave. Rel. Error | | | | Max. Rel. Error | | | |
|---|---|---|---|---|---|---|---|---|
| Predefined Rel. Subopt. | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ |
| $ny = 10$ | 0.02 | 0.02 | 0.04 | 0.08 | 0.96 | 0.98 | 0.97 | 0.96 |
| $ny = 20$ | 0.01 | 0.02 | 0.09 | 0.02 | 0.57 | 0.63 | 0.86 | 0.94 |
| $ny = 50$ | 0.01 | 0.05 | 0.19 | 0.35 | 0.80 | 0.91 | 0.98 | 0.89 |
| $ny = 100$ | 0.03 | 0.09 | 0.26 | 0.38 | 0.74 | 0.89 | 0.95 | 0.83 |
| $ny = 200$ | 0.01 | 0.09 | 0.25 | 0.39 | 0.72 | 0.73 | 0.86 | 0.71 |

relative suboptimality is 0.0424, which is a sufficient condition for ASBSA to be terminated within finite iterations. In fact, it is quite reasonable to take much smaller $\epsilon$ in practice, e.g. in the next subsection, predefined relative suboptimality is set as small as 0.0001, and all the tests can generate $\epsilon$ suboptimal solution within in finite iterations.

### 5.5.2 Random tests between POASIM and ASBSA

In this subsection, for each $ny$ equals $10, 20, 50, 100$, and 200, 1000 independent randomly generated linear constrained quadratic problems are used to test POASIM, each of

Table 5.3 – Random tests statistics of POASIM: active inequality constraints ratio and number of LP calculated

| Problem size | Active Inequality Constraints Ratio | | | Number of LP Calculated | | |
|---|---|---|---|---|---|---|
| | Ave. | Max. | Min. | Ave. | Max. | Min. |
| $ny = 10$ | 12.64% | 25.00% | 5.00% | 3.76 | 11 | 1 |
| $ny = 20$ | 12.45% | 25.00% | 2.50% | 6.59 | 19 | 1 |
| $ny = 50$ | 13.53% | 25.00% | 5.00% | 15.77 | 41 | 5 |
| $ny = 100$ | 14.16% | 24.50% | 6.00% | 28.19 | 73 | 11 |
| $ny = 200$ | 14.92% | 24.25% | 8.50% | 49.94 | 127 | 23 |

Table 5.4 – Random tests statistics of ABSBA: number of $\boldsymbol{y}^*_{\mathcal{A}^k}$ returned and CP calculated

| $ny$ | | 10 | 20 | 50 | 100 | 200 |
|---|---|---|---|---|---|---|
| $10^{-4}$ | Ave. CP | 0.07 | 0.04 | 0.09 | 0.24 | 0.40 |
| | Max. CP | 2 | 2 | 3 | 4 | 7 |
| | Min. CP | 0 | 0 | 0 | 0 | 0 |
| | $\boldsymbol{y}^*_{\mathcal{A}^k}$ | 937 | 975 | 941 | 849 | 728 |
| $10^{-3}$ | Ave. CP | 0.11 | 0.12 | 0.27 | 0.55 | 0.84 |
| | Max. CP | 3 | 2 | 6 | 5 | 6 |
| | Min. CP | 0 | 0 | 0 | 0 | 0 |
| | $\boldsymbol{y}^*_{\mathcal{A}^k}$ | 915 | 933 | 821 | 618 | 402 |
| $10^{-2}$ | Ave. CP | 0.29 | 0.42 | 1.03 | 1.55 | 2.06 |
| | Max. CP | 3 | 5 | 9 | 11 | 21 |
| | Min. CP | 0 | 0 | 0 | 0 | 0 |
| | $\boldsymbol{y}^*_{\mathcal{A}^k}$ | 843 | 773 | 413 | 123 | 17 |
| $10^{-1}$ | Ave. CP | 0.73 | 1.29 | 2.58 | 3.92 | 6.36 |
| | Max. CP | 6 | 11 | 23 | 44 | 86 |
| | Min. CP | 0 | 0 | 0 | 0 | 1 |
| | $\boldsymbol{y}^*_{\mathcal{A}^k}$ | 671 | 422 | 53 | 4 | 0 |

Table 5.5 – Random tests statistics: computation time ratio of one CP in ABSBA to one LP in POASIM

| $ny$ | | 10 | 20 | 50 | 100 | 200 |
|---|---|---|---|---|---|---|
| $10^{-4}$ | Ave. | 0.88 | 1.10 | 2.09 | 3.07 | 3.40 |
| | Max. | 1.91 | 2.85 | 6.16 | 8.17 | 8.63 |
| | Min. | 0.22 | 0.56 | 0.79 | 1.29 | 1.36 |
| $10^{-3}$ | Ave. | 0.86 | 1.19 | 2.41 | 3.58 | 3.79 |
| | Max. | 1.71 | 3.47 | 8.00 | 10.43 | 10.70 |
| | Min. | 0.21 | 0.53 | 0.78 | 1.17 | 1.37 |
| $10^{-2}$ | Ave. | 0.84 | 1.40 | 2.87 | 4.15 | 4.53 |
| | Max. | 2.09 | 4.01 | 9.15 | 10.55 | 10.65 |
| | Min. | 0.22 | 0.48 | 0.71 | 1.24 | 1.62 |
| $10^{-1}$ | Ave. | 0.86 | 1.37 | 2.85 | 4.14 | 3.99 |
| | Max. | 2.27 | 4.29 | 9.01 | 11.30 | 12.21 |
| | Min. | 0.21 | 0.48 | 0.79 | 1.84 | 2.21 |

which is tested under 4 different relative suboptimality [6] particularly for ASBSA: 0.0001, 0.001, 0.01 and 0.1.

Table 5.2 shows that the predefined suboptimality of all random tests is fulfilled, the maximum relative error in general is significantly larger than the average relative error of all tests. What is not presented in Table 5.2 but worth mentioning is that all solutions delivered by ASBSA are primal feasible.

For every figure from Fig. 5.1 to 5.5, as predefined relative suboptimality increases from $1 \times 10^{-4}$ to $1 \times 10^{-1}$, the boxplot of iteration number ratio of ASBSA to POASIM declines steadily. The reason behind this is that the higher suboptimality, the higher tolerance of incorrectness of $\mathcal{A}^k$, thus the higher possibility for (5.15) or (5.16) to occur, since the gap between $\tilde{g}^k_{best}$ and $p^*$ becomes more tolerated. This tendency is also shown in Table. 5.4, where the average number of CP calculated grows exponentially as predefined relative suboptimality increases for all $ny$ cases. Regarding the influence of problem size, under each suboptimality, the iteration number ratio of ASBSA to POASIM decreases remarkably as $ny$ increases, primarily due to 2 factors shown in Table. 5.3: first, the number of LP calculated grows proportionally to $ny$; second, comparable low number (less than 10) of LP for $ny$ equals 10 and 20, while ASBSA needs at least several tens of iterations to make (5.15) or (5.16) happen. As a result, ASBSA only comes into statistically dominant w.r.t. iteration number starting from $ny = 50$ under predefined relative suboptimality

6. The relative suboptimality is computed as suboptimality divided by $p^*$.

$10^{-1}$, and achieves the best performance in case $ny = 200$ with overall superiority under $10^{-1}$ and $10^{-2}$.

Few facts for time performance analysis: computing CP dominates the total computation time of ASBSA, as it generally consumes $4 - 5 \times 10^4$ fold time than that of one iteration of (5.30); computation time ratio of one CP to one LP increases logarithmically as $ny$ increases under the same predefined relative suboptimality, and the ratio is comparably insensitive to the change of suboptimality. As shown from Fig. 5.6 to 5.10, ASBSA outperforms POASIM in terms of computation time under relative suboptimality $1 \times 10^{-4}$, $1 \times 10^{-3}$ and $1 \times 10^{-2}$ in all $ny$ cases, for comparably low average number of CP calculated for ASBSA in these cases (showed in Table. 5.4). Note that the considerable number of $\boldsymbol{y}^*_{\mathcal{A}^k}$ returned as the primal suboptimal solution (showed in Table. 5.4) can also account for time supremacy of ASBSA, which spares the effort in computing CP and results in even less computation time.

## 5.6 Conclusion

In this chapter, combining the first order gradient method and KKT criterion, a proactive method (POASIM) has been proposed in solving for the optimal solution for linear constrained quadratic optimization by dynamically identifying the active set in a iterative manner. In the hope of terminating the process faster, a suboptimal method (ASBSA) based on cone programming has been further initiated to generate suboptimal and feasible solutions. The suboptimal method can be considerably beneficial when the optimal active set is prohibitive to identify during the iterative process. A lower bound of suboptimality has been demonstrated, above which ASBSA can generate suboptimal solutions within finite iterations with no information nor assumption on the optimal active set.

Through random numerical experiments, the $\epsilon$-suboptimality and feasibility have been verified for the suboptimal method, which has moreover revealed statistical improvement of computation time and iteration number compared to the proactive method under certain predefined relative suboptimality and problem sizes.

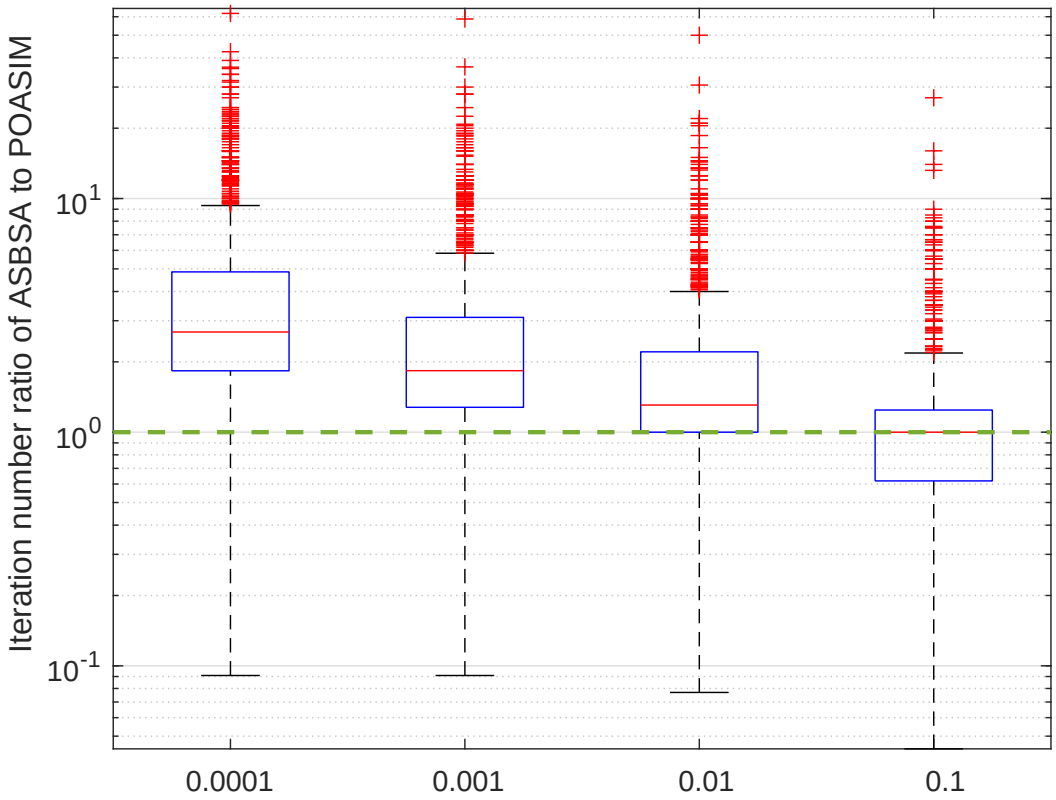


Figure 5.1 – Iteration number ratio of ASBSA to POASIM with $ny = 10$
Sample value exceeded $+/-2.7\sigma$ shows as whisker, the same setting for other box plots. Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes fewer, more, or the same iterations as POASIM in the same test. The lower value, the better performance of ASBSA.

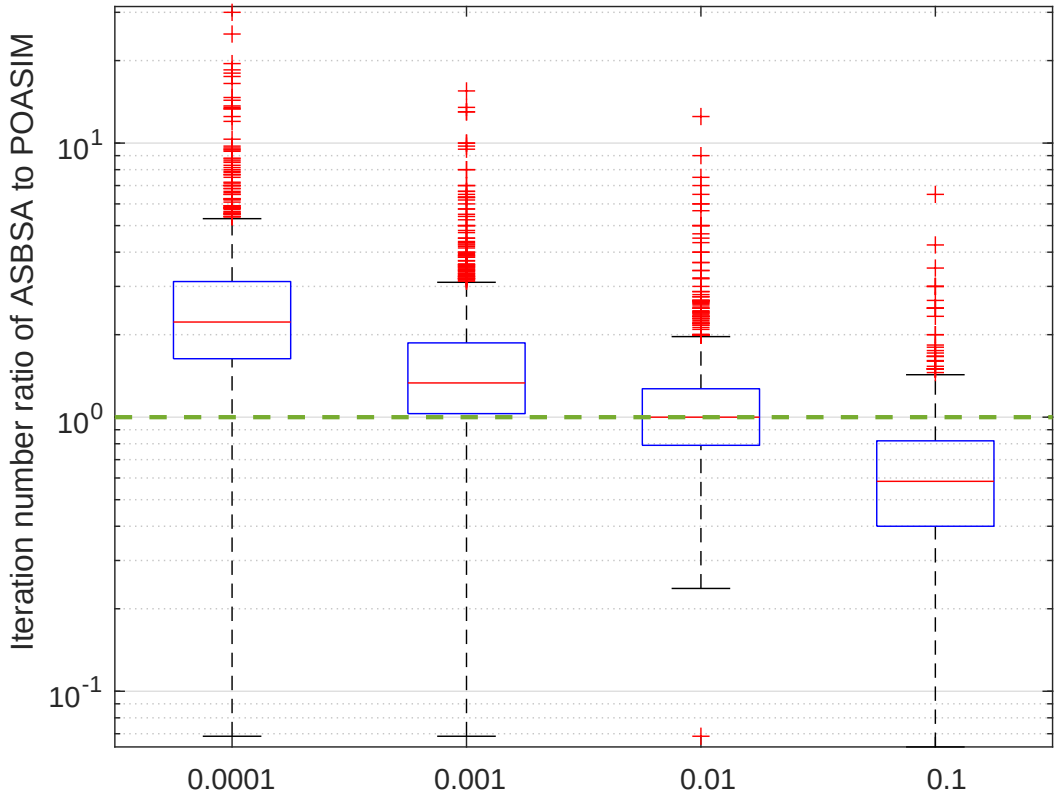


Figure 5.2 – Iteration number ratio of ASBSA to POASIM with $ny = 10$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes fewer, more, or the same iterations as POASIM in the same test. The lower value, the better performance of ASBSA.

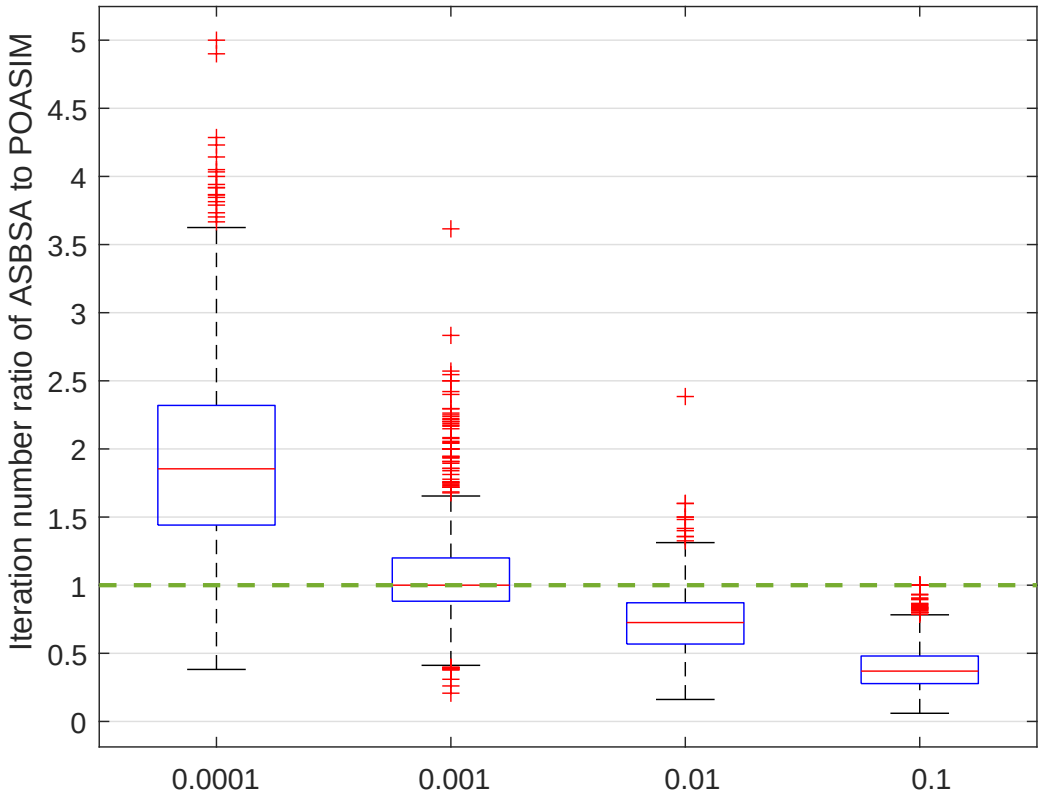


Figure 5.3 – Iteration number ratio of ASBSA to POASIM with $ny = 10$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes fewer, more, or the same iterations as POASIM in the same test. The lower value, the better performance of ASBSA.

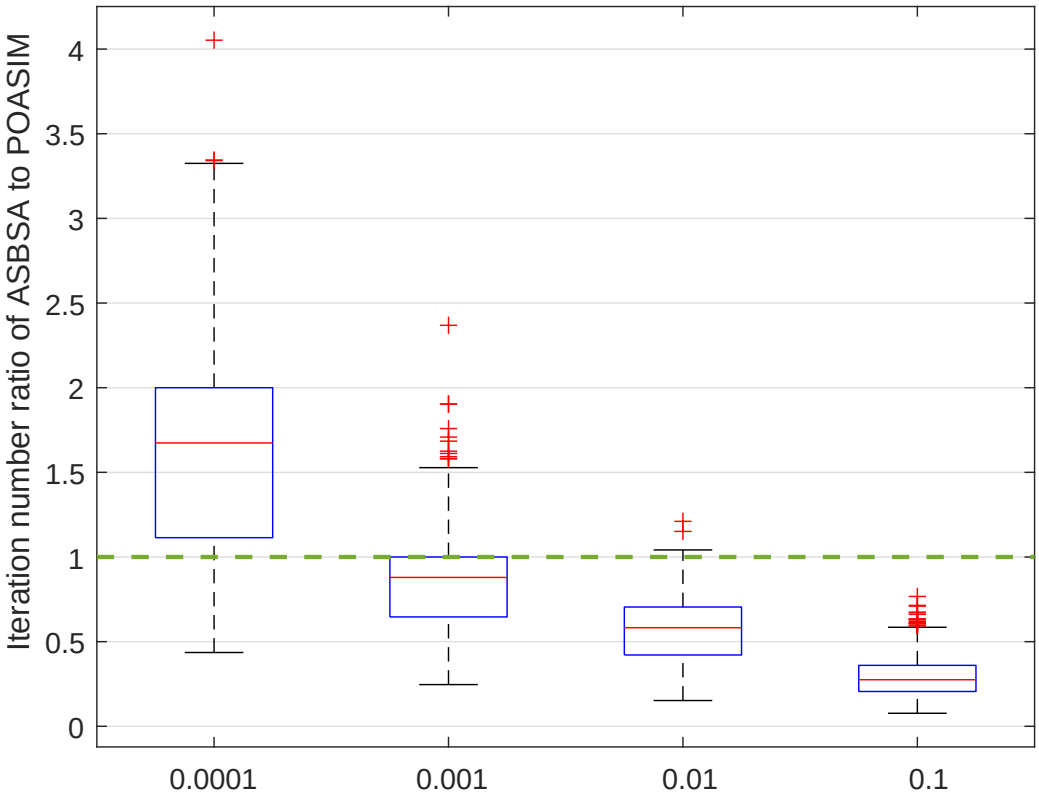


Figure 5.4 – Iteration number ratio of ASBSA to POASIM with $ny = 10$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes fewer, more, or the same iterations as POASIM in the same test. The lower value, the better performance of ASBSA.

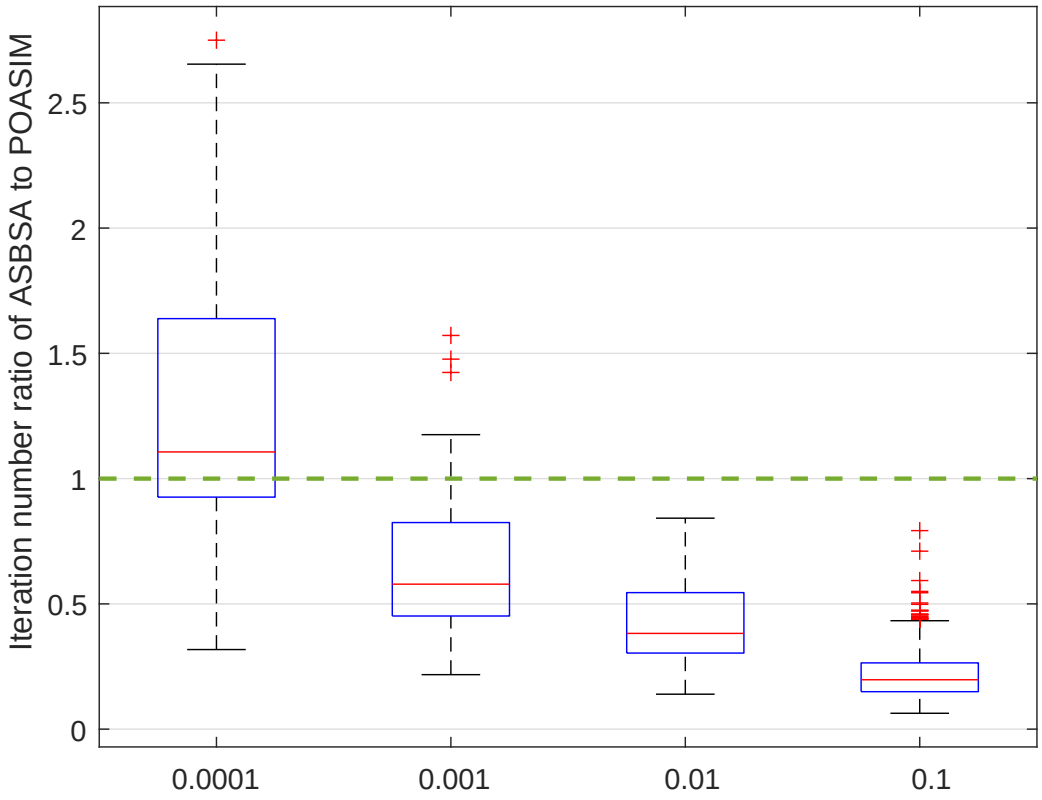


Figure 5.5 – Iteration number ratio of ASBSA to POASIM with $ny = 10$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes fewer, more, or the same iterations as POASIM in the same test. The lower value, the better performance of ASBSA.

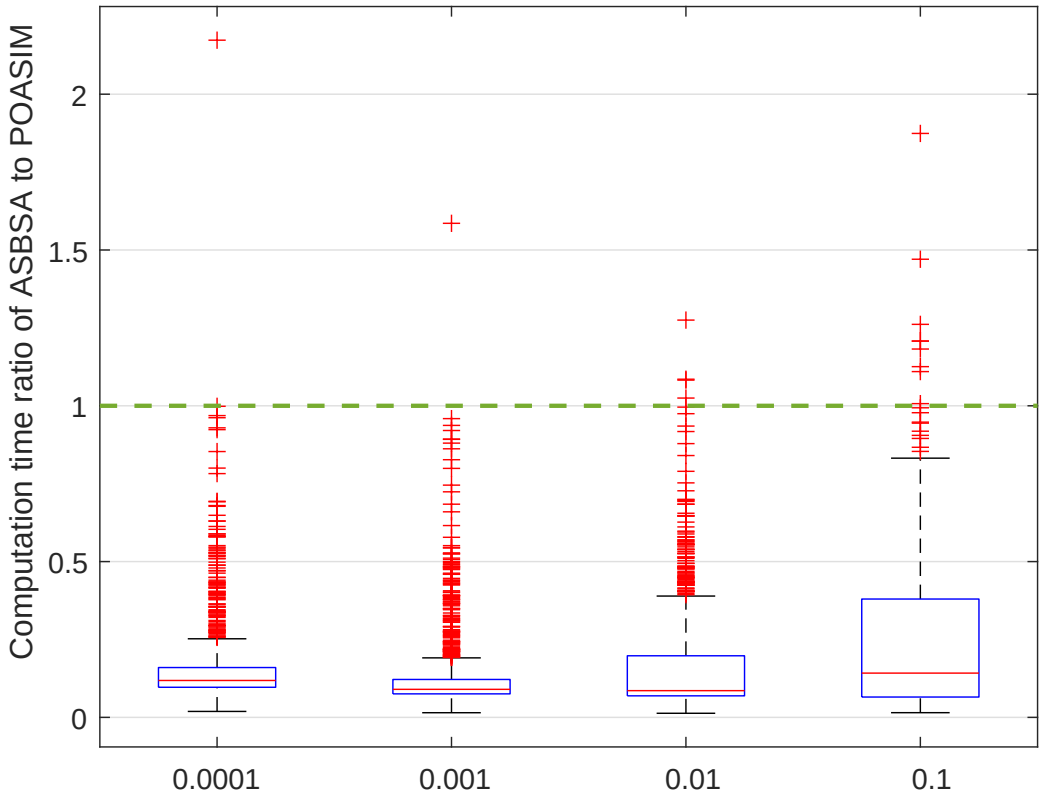


Figure 5.6 – Computation time ratio of ASBSA to POASIM with $ny = 10$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes less, more, or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.

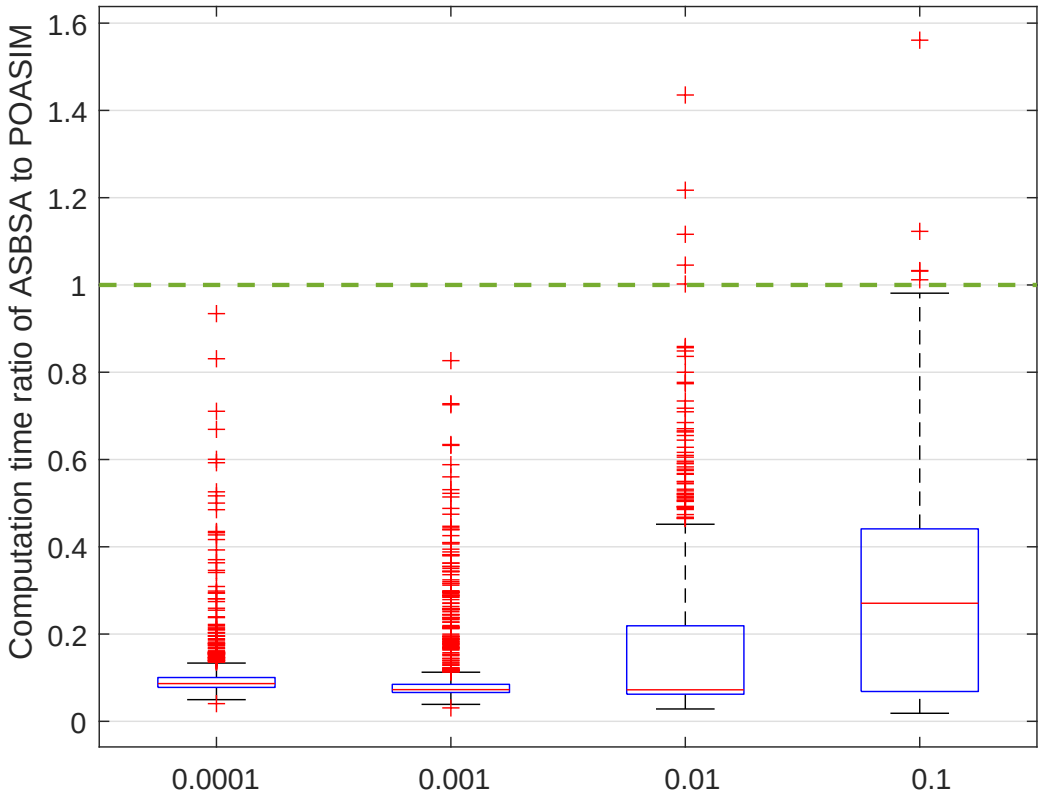


Figure 5.7 – Computation time ratio of ASBSA to POASIM with $ny = 20$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes less, more, or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.

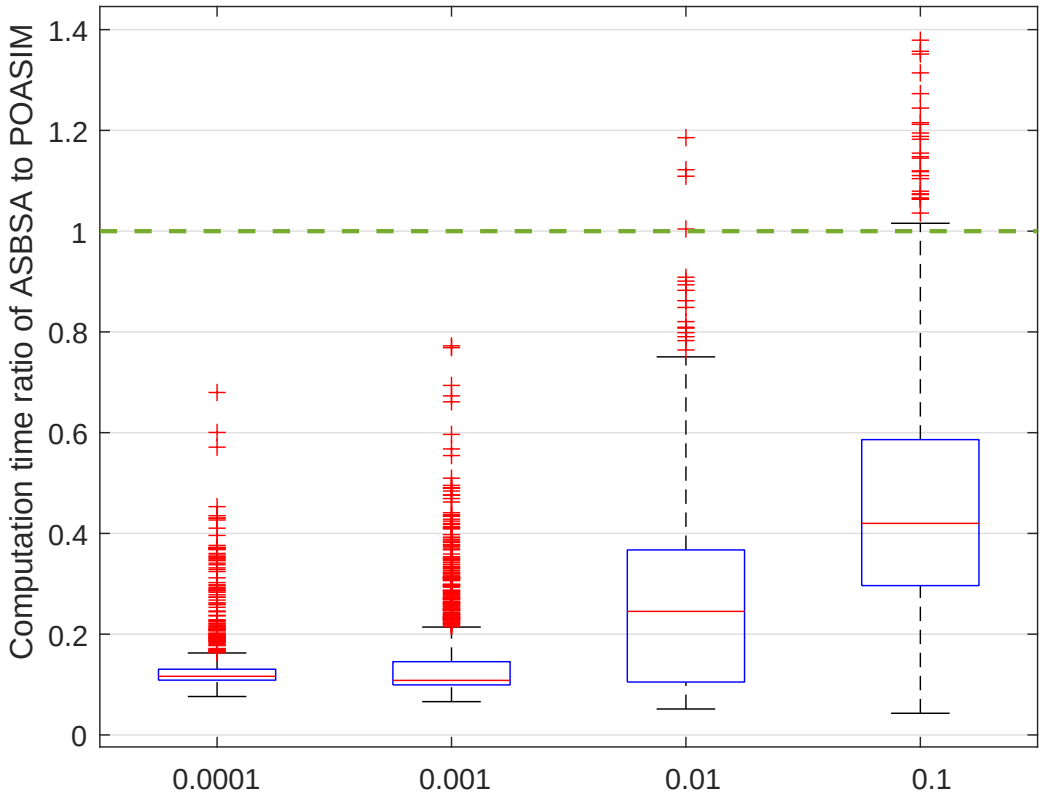


Figure 5.8 – Computation time ratio of ASBSA to POASIM with $ny = 50$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes less, more, or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.

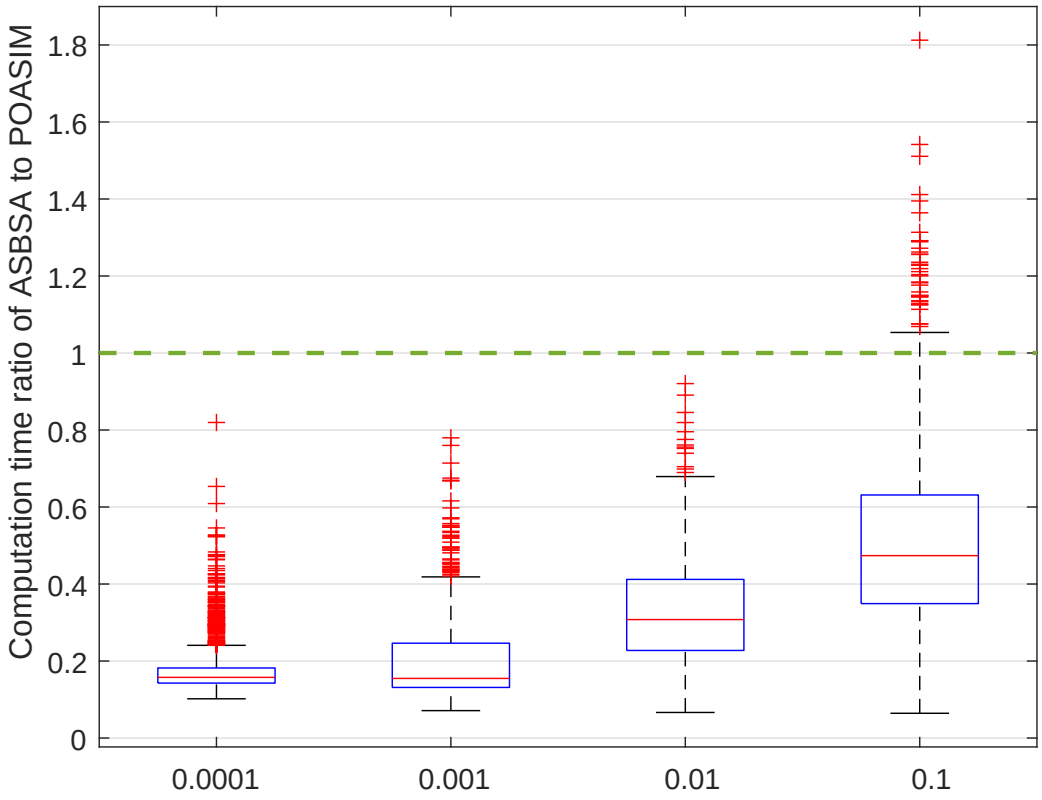


Figure 5.9 – Computation time ratio of ASBSA to POASIM with $ny = 100$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes less, more, or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.

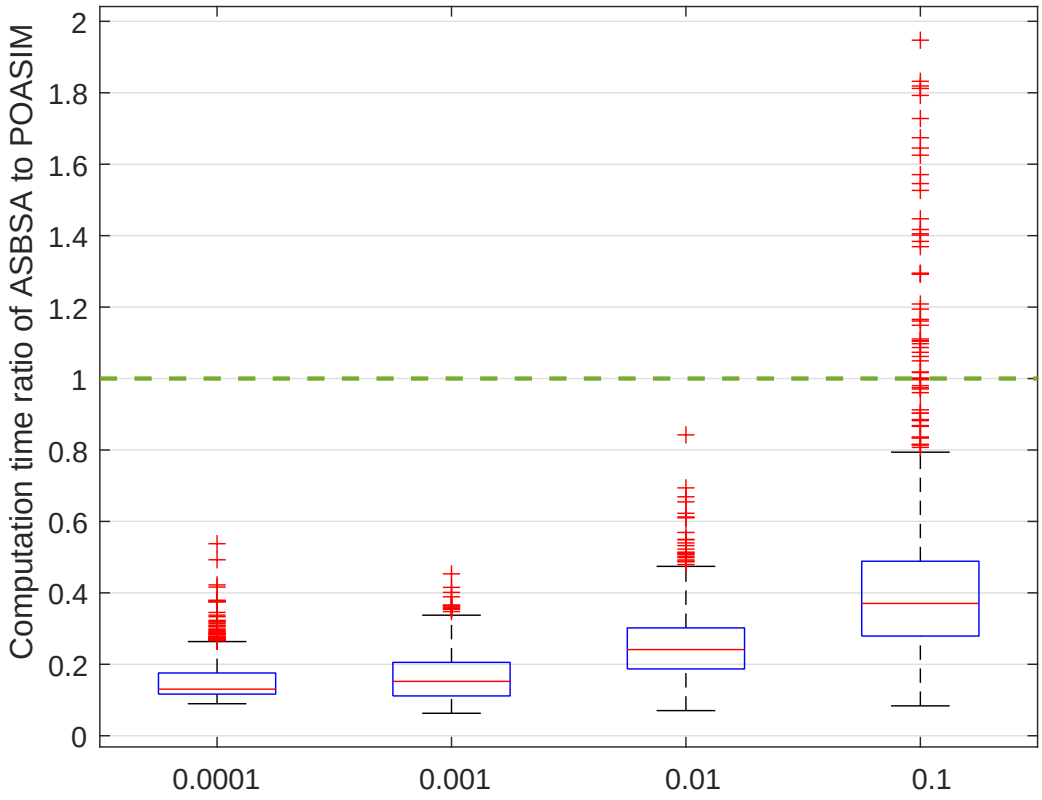


Figure 5.10 – Computation time ratio of ASBSA to POASIM with $ny = 200$
Sample value less, greater than, or equals to 1 (green horizontal line) means ASBSA consumes less, more, or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.

Chapter 6

# Conclusion

## 6.1 Conclusion

In this dissertation, targeting at accelerating the iterative solving process of MPC resulted or related convex optimization problem, several algorithms have been proposed, each of which has been studied theoretically and experimentally in comparison with the referred conventional approach.

### 6.1.1 Conclusion of algorithms and methods proposed

The overall conclusion of each algorithm proposed in this dissertation is given below.

Under the step-wise stopping condition in a distributed MPC, 2 algorithms have been proposed in §2, which can reduce the iterations requirement by fixing the Lagrange multipliers' value and dropping satisfying steps in the MPC problem respectively. In the numerical experiment, through dynamically reducing problem size and using a "warm start" strategy, the dynamic sizing algorithm has consumed essentially fewer iteration number and less computation time compared to Uzawa method in meeting the same stopping condition.

In the setting of equality constrained distributed MPC with coupling dynamics, the full prediction horizon projection algorithm proposed in §3 has employed an integrated gradient-based criterion and projection mechanism to guarantee suboptimality and feasibility. In exploiting the MPC feature that only the first step of control sequence is applied, a first step focused projection algorithm with suboptimality and feasibility guarantee has been proposed in §3, which can substantially reduce the iteration number and computation time in meeting the same suboptimality requirement. Its effectiveness has been verified through random numerical experiments, in which the advantages of iteration number and computation time has continued to grow as required suboptimality decreases.

In MPC setting with coupling dynamics and step separable convex inequality con-

straints, the first step focused criterion in meeting with suboptimality under bounded infeasibility has been proposed in §4. The resulting algorithm has been demonstrated to be superior than the traditional primal dual interior point method in iteration number both theoretically and experimentally. In terms of computation time, it has shown a statistical advantage in medium and long prediction horizon cases under all tested suboptimality compared to the traditional primal dual interior point method.

For general linear constrained quadratic optimization (not necessarily be MPC), combining active set identification technique and general gradient method, a proactive method has been proposed to deliver the optimal solution in §5. In addition, a suboptimal algorithm has been proposed in §5 based on cone programming to accelerate the iterative process termination in generating feasible solutions with guaranteed suboptimality. The lower bound of suboptimality has also been demonstrated, above which the suboptimality can lead to the termination of iterative process within finite iterations. Through random numerical experiments, the suboptimal algorithm has statistically outperformed the proactive method in iteration number under low suboptimality and large problem sizes, and in computation time under low and medium suboptimality with all problem sizes.

### 6.1.2 Comparison among algorithms and methods proposed

Of all the algorithms proposed in this dissertation, the comparison of specific constraints tackled, iterative manner structure, and iterative method type is exhibited in Table. 6.1. In parallel, for performance overview, the comparison including features of proposed algorithms on convergence rate, with or without $\epsilon$ suboptimality guarantee, primal feasible or not, with or without iteration number guarantee and solution length is summarized in Table. 6.2. These 2 tables can be utilized to match the specific computation target and problem setting to the suited algorithm, thus forming a handy guideline for study and application.

## 6.2 Future directions

Future work with regard to algorithms and techniques proposed in this dissertation can be addressed in the following directions:

1. expand application of techniques proposed:

- application of step drop treatment and "warm start" strategy (§2) under quantified criteria and recursive implementation,
- employment of gradient based suboptimality condition and feasibility guaranteed projection (§3) in linear inequality constrained cases, which could be either eliminated by barrier function [8], or transformed into equality constraints by active constraint identification technique[43] [44] or active set method[65],
- application of the first step focused criterion to the primal dual interior point method (§4) with no additional assumption on dual residual, or no presence of dual residual,
- application of cone programming, backtracking mechanism during the iterative process, approximation of optimal objective value (§5) in MPC context, and in various problem sizes and settings,
- verification of effectiveness and efficiency of algorithms and methods proposed through real case study and large scale systems;

2. study properties of algorithms proposed in control and optimization theory:
   - stability proof establishment of suboptimal algorithms proposed for MPC, with or without terminal state constraints,
   - recursive feasibility verification,
   - implementation of primal dual interior point method using Hessian approximation/factorization with banded or sparse structure, and quasi-Newton method,
   - complexity study of algorithms proposed to determine the suited circumstances with theoretical proofs;
3. combine techniques proposed with other methods:
   - combination of the accelerated termination idea with machine learning for better performance,
   - combination of accelerated termination idea with matured optimization techniques, e.g., semidefinite programming, alternative direction multiplier method, augmented Lagrangian, etc.

Table 6.1 – Problem and optimization method oriented indicators of algorithms proposed

| Alg. | Location | Iteration method | Inequality Cons. type | Inequality Cons. requirement | Inequality Cons. structure | Iteration structure | Equality Cons. (Dynamics) |
|---|---|---|---|---|---|---|---|
| DLMFA | §2.4.1 | gradient | general convex | continuous | subsystem & step-wise separable | distributed | uncoupled |
| LOPDSA | §2.4.2 | ditto* | ditto | ditto | ditto | ditto | ditto |
| FPH-P | §3.3.2 | Nesterov descent | not considered | not applicable | not applicable | distributed | coupling |
| FS-P | §3.4.2 | ditto | ditto | ditto | ditto | ditto | ditto |
| ATPDIPM | §4.4.2 | Primal dual interior point method | general convex | twice differentiable | step-wise separable | centralized | coupling |
| POASIM | §5.3 | general gradient | linear | not applicable | general | centralized | general linear |
| ASBSA | §5.4.2 | ditto | ditto | ditto | ditto | ditto | ditto |

* ditto means that the content in the current cell is the same as that of the neighbor cell above in the same column, and the same setting is applied to Table. 6.2.

Table 6.2 – Performance oriented indicators of algorithms proposed

| Alg. | Location | Convergence rate | $\epsilon$ suboptimality guarantee | Primal feasibility | Solution length | Iteration number guarantee |
|---|---|---|---|---|---|---|
| DLMFA | §2.4.1 | $\mathcal{O}(1/k)$ | no | no | full sequence | - |
| LOPDSA | §2.4.2 | ditto | ditto | ditto | ditto | ditto |
| FPH-P | §3.3.2 | $\mathcal{O}(1/k^2)$ | yes | yes | full sequence | - |
| FS-P | §3.4.2 | ditto | ditto | ditto | first step | $\leq$ FPH-P |
| ATPDIPM | §4.4.2 | quadratic | yes | bounded | first step | $\leq$ PDIPM |
| POASIM | §5.3 | not applicable | yes | yes | full sequence | - |
| ASBSA | §5.4.2 | ditto | ditto | ditto | ditto | finite* |

* For $\epsilon$ greater than the lower bound, an $\epsilon$ primal suboptimal solution can be solved within finite iteration, see details in §5.4.3.

# APPENDICES

## A Parameters setting of test in §4.5.1

For $l = 1, ..., 5$, the initial state $\bar{x}_1$ is listed as follows:

$$\bar{x}_1 = \begin{bmatrix} -0.0206 \\ -0.3389 \end{bmatrix}, \ \bar{x}_2 = \begin{bmatrix} -0.0227 \\ -0.3715 \end{bmatrix}, \ \bar{x}_3 = \begin{bmatrix} -0.2808 \\ -0.1042 \end{bmatrix},$$
$$\bar{x}_4 = \begin{bmatrix} 0.1813 \\ 0.1204 \end{bmatrix}, \ \bar{x}_5 = \begin{bmatrix} 0.4529 \\ -0.4253 \end{bmatrix}.$$

For $l = 1, ..., 5$ and $i = 1, ..., 5$, $A_{li}$ is listed as follows:

$$A_{11} = \begin{bmatrix} 0.0022 & -0.0088 \\ -0.0088 & 0.0178 \end{bmatrix}, \ A_{12} = \begin{bmatrix} -0.9526 & 0.0894 \\ 0.0894 & -0.4309 \end{bmatrix},$$
$$A_{13} = \begin{bmatrix} -0.7201 & -0.0294 \\ -0.0294 & -0.7301 \end{bmatrix}, \ A_{14} = \begin{bmatrix} 0.3288 & -0.8206 \\ 0.8206 & 0.3288 \end{bmatrix},$$
$$A_{15} = \begin{bmatrix} -0.1094 & 0.5314 \\ 0.5314 & 0.0895 \end{bmatrix}, \ A_{21} = \begin{bmatrix} 0.4027 & -0.1144 \\ -0.1144 & 0.2278 \end{bmatrix},$$
$$A_{22} = \begin{bmatrix} -0.4210 & -0.1690 \\ -0.1690 & -0.5903 \end{bmatrix}, \ A_{23} = \begin{bmatrix} -0.1761 & 0.1235 \\ 0.1235 & -0.1211 \end{bmatrix},$$
$$A_{24} = \begin{bmatrix} 0.3077 & -0.1710 \\ 0.1710 & 0.3077 \end{bmatrix}, \ A_{25} = \begin{bmatrix} -0.5149 & 0.0562 \\ 0.0562 & -0.5087 \end{bmatrix},$$
$$A_{31} = \begin{bmatrix} -0.5770 & 0.2454 \\ 0.2454 & -0.5099 \end{bmatrix}, \ A_{32} = \begin{bmatrix} -0.4570 & 0.2825 \\ 0.2825 & -0.7817 \end{bmatrix},$$
$$A_{33} = \begin{bmatrix} -0.0161 & -0.1895 \\ 0.1895 & -0.0161 \end{bmatrix}, \ A_{34} = \begin{bmatrix} 0.3047 & -0.2041 \\ -0.2041 & 0.8310 \end{bmatrix},$$
$$A_{35} = \begin{bmatrix} 0.2651 & 0.1833 \\ 0.1833 & 0.2843 \end{bmatrix}, \ A_{41} = \begin{bmatrix} 0.9479 & 0.2706 \\ 0.2706 & -0.4063 \end{bmatrix},$$

$$A_{42} = \begin{bmatrix} 0.9224 & 0.0638 \\ 0.0638 & 0.9476 \end{bmatrix}, \; A_{43} = \begin{bmatrix} 0.4919 & 0.5599 \\ 0.5599 & 0.3918 \end{bmatrix},$$
$$A_{44} = \begin{bmatrix} 0.0319 & 0.0848 \\ 0.0848 & 0.6632 \end{bmatrix}, \; A_{45} = \begin{bmatrix} 0.5160 & 0.3943 \\ 0.3943 & 0.3281 \end{bmatrix},$$
$$A_{51} = \begin{bmatrix} 0.7582 & -0.2475 \\ -0.2475 & 0.5570 \end{bmatrix}, \; A_{52} = \begin{bmatrix} 0.1910 & 0.8040 \\ 0.8040 & 0.2009 \end{bmatrix},$$
$$A_{53} = \begin{bmatrix} 0.7958 & -0.3538 \\ -0.3538 & 0.0994 \end{bmatrix}, \; A_{54} = \begin{bmatrix} 0.7921 & -0.1207 \\ -0.1207 & 0.9295 \end{bmatrix},$$
$$A_{55} = \begin{bmatrix} 0.3815 & -0.1060 \\ -0.1060 & 0.6187 \end{bmatrix}.$$

For $l = 1, ..., 5$ and $i = 1, ..., 5$, $B_{li}$ is listed as follows:

$$B_{11} = \begin{bmatrix} 0.5721 & 0.38373 \\ 0.4684 & -1.1608 \end{bmatrix}, \; B_{12} = \begin{bmatrix} -0.8422 & 0.7102 \\ 0 & 1.8255 \end{bmatrix},$$
$$B_{13} = \begin{bmatrix} -0.8236 & -1.0633 \\ 1.4745 & 0 \end{bmatrix}, \; B_{14} = \begin{bmatrix} 0.5568 & -1.1780 \\ 0 & -0.6155 \end{bmatrix},$$
$$B_{15} = \begin{bmatrix} 0.6988 & -0.2987 \\ 0 & 0 \end{bmatrix}, \; B_{21} = \begin{bmatrix} 1.1273 & 0.7249 \\ 0 & 1.2263 \end{bmatrix},$$
$$B_{22} = \begin{bmatrix} -1.06116 & -1.1690 \\ -0.2915 & 0 \end{bmatrix}, \; B_{23} = \begin{bmatrix} 0.5182 & 0.0458 \\ 0 & -0.2173 \end{bmatrix},$$
$$B_{24} = \begin{bmatrix} 1.2777 & -0.2188 \\ 0 & 0 \end{bmatrix}, \; B_{25} = \begin{bmatrix} 0 & 0 \\ 1.4087 & -1.0924 \end{bmatrix},$$
$$B_{31} = \begin{bmatrix} 0.2019 & 1.3625 \\ 0.8934 & -0.7657 \end{bmatrix}, \; B_{32} = \begin{bmatrix} 0 & -1.5257 \\ -0.0227 & 0 \end{bmatrix},$$
$$B_{33} = \begin{bmatrix} 0 & -1.4667 \\ 0 & 0 \end{bmatrix}, \; B_{34} = \begin{bmatrix} 2.4911 & 0 \\ -0.2947 & 0.4339 \end{bmatrix},$$
$$B_{35} = \begin{bmatrix} 0.0162 & 0.3014 \\ 0 & 0 \end{bmatrix}, \; B_{41} = \begin{bmatrix} 0 & -1.8177 \\ -0.8237 & -1.0549 \end{bmatrix},$$
$$B_{42} = \begin{bmatrix} -0.8770 & 0.4858 \\ -0.1062 & 0.4752 \end{bmatrix}, \; B_{43} = \begin{bmatrix} -0.2659 & 0 \\ 0 & -0.2041 \end{bmatrix},$$

$$B_{44} = \begin{bmatrix} 0 & 0 \\ 1.5776 & -0.1874 \end{bmatrix}, \; B_{45} = \begin{bmatrix} 0 & 1.1115 \\ 0.5311 & 0.5826 \end{bmatrix},$$
$$B_{51} = \begin{bmatrix} 1.3649 & 0.0066 \\ 0.2663 & -0.4058 \end{bmatrix}, \; B_{52} = \begin{bmatrix} 0.6295 & 0 \\ -0.6692 & 0 \end{bmatrix},$$
$$B_{53} = \begin{bmatrix} 0.3811 & 1.7053 \\ -0.4386 & 0 \end{bmatrix}, \; B_{54} = \begin{bmatrix} -0.3698 & 1.4027 \\ -0.4564 & 1.0883 \end{bmatrix},$$
$$B_{55} = \begin{bmatrix} 0.0800 & -0.7723 \\ 0 & 0.3162 \end{bmatrix}.$$

# B Parameters setting of test in §5.5.1

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0.2410 & 0.0000 & 0.0000 & 0.3928 & -0.4095 \\ 0.2647 & 0.0368 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.1215 & 0.0000 & -0.1425 \\ 0.0000 & 0.0000 & -0.2545 & 0.0000 & -0.1178 \\ 0.0000 & -0.4767 & -0.4422 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.4789 & -0.4363 & 0.2437 \\ -0.3338 & 0.1515 & -0.0908 & -0.0650 & 0.0000 \\ 0.0000 & 0.0000 & 0.4786 & -0.4162 & 0.0000 \\ 0.0368 & 0.3294 & 0.0000 & 0.0000 & 0.4502 \\ 0.0000 & 0.0000 & 0.0457 & 0.0000 & 0.1763 \end{bmatrix}^T,$$

$$\tilde{\boldsymbol{b}} = \begin{bmatrix} 0.1904 & 0.0343 & -0.6409 & -0.3417 & 0.0641 \end{bmatrix}^T.$$

# BIBLIOGRAPHY


[1] Alessandro Alessio and Alberto Bemporad, « A Survey on Explicit Model Predictive Control », *in*: *Nonlinear Model Predictive Control*, Springer, 2009, pp. 345–369.

[2] Jonathan Barzilai and Jonathan M Borwein, « Two-Point Step Size Gradient Methods », *in*: *IMA journal of numerical analysis* 8.*1* (1988), pp. 141–148, DOI: 10.1093/imanum/8.1.141.

[3] Amir Beck and Marc Teboulle, « A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems », *in*: *SIAM journal on imaging sciences* 2.*1* (2009), pp. 183–202, DOI: 10.1137/080716542.

[4] Alberto Bemporad and Carlo Filippi, « Suboptimal Explicit Rhc via Approximate Multiparametric Quadratic Programming », *in*: *Technical Report ETH Zurich, AUT02-07* (2002).

[5] Alberto Bemporad et al., « The Explicit Linear Quadratic Regulator for Constrained Systems », en, *in*: *Automatica* 38.*1* (Jan. 2002), pp. 3–20, ISSN: 0005-1098, DOI: 10.1016/S0005-1098(01)00174-1.

[6] Dimitri P Bertsekas, *Nonlinear Programming*, Athena scientific Belmont, 1999.

[7] Stephen Boyd et al., « Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers », *in*: *Foundations and Trends® in Machine learning* 3.*1* (2011), pp. 1–122, DOI: 10.1561/2200000016.

[8] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[9] Philipp Braun et al., « Hierarchical Distributed ADMM for Predictive Control with Applications in Power Networks », *in*: *IFAC Journal of Systems and Control* 3 (2018), pp. 10–22, DOI: 10.1016/j.ifacsc.2018.01.001.

[10] Eduardo F. Camacho and Carlos Bordons, « Nonlinear Model Predictive Control: An Introductory Review », *in*: *Assessment and Future Directions of Nonlinear Model Predictive Control*, Springer, 2007, pp. 1–16.

[11] Eduardo F Camacho and Carlos Bordons Alba, *Model Predictive Control*, Springer Science & Business Media, 2013.

[12] Y. Censor et al., « On the Effectiveness of Projection Methods for Convex Feasibility Problems with Linear Inequality Constraints », en, *in*: *arXiv:0912.4367 [math]* (Dec. 2009), arXiv: 0912.4367 [math].

[13] Mung Chiang et al., « Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures », *in*: *Proceedings of the IEEE* 95.*1* (2007), pp. 255–312.

[14] Eunkyoung G. Cho et al., « Rolling Horizon Scheduling of Multi-Factory Supply Chains », *in*: *Winter Simulation Conference*, vol. 2, 2003, pp. 1409–1416.

[15] David W. Clarke, Coorous Mohtadi, and P. S. Tuffs, « Generalized Predictive Control—Part I. The Basic Algorithm », *in*: *Automatica* 23.*2* (1987), pp. 137–148.

[16] Charles R. Cutler and Brian L. Ramaker, « Dynamic Matrix Control?? A Computer Control Algorithm », *in*: *Joint Automatic Control Conference*, 17, 1980, p. 72.

[17] Xiang Dai, Romain Bourdais, and Hervé Guéguen, « Dynamic Reduction of the Iterations Requirement in a Distributed Model Predictive Control », *in*: *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 6392–6397, DOI: 10.1109/CDC40024.2019.9029783.

[18] Herbert Dawid, « Long Horizon versus Short Horizon Planning in Dynamic Optimization Problems with Incomplete Information », *in*: *Economic Theory* 25.*3* (2005), pp. 575–597, DOI: 10.1007/s00199-003-0437-5.

[19] Alexander Domahidi et al., « Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control », *in*: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 668–674, DOI: 10.1109/CDC.2012.6426855.

[20] Marguerite Frank and Philip Wolfe, « An Algorithm for Quadratic Programming », *in*: *Naval research logistics quarterly* 3.*1-2* (1956), pp. 95–110, DOI: 10.1002/nav.3800030109.

[21] Fabio Furini et al., « QPLIB: A Library of Quadratic Programming Instances », *in*: *Mathematical Programming Computation* 11.*2* (2019), DOI: 10.1007/s12532-018-0147-4.

[22] Carlos E. Garcia, David M. Prett, and Manfred Morari, « Model Predictive Control: Theory and Practice—A Survey », *in*: *Automatica* 25.*3* (1989), pp. 335–348, DOI: 10.1016/0005-1098(89)90002-2.

[23] Pontus Giselsson and Anders Rantzer, « On Feasibility, Stability and Performance in Distributed Model Predictive Control », *in*: *IEEE Transactions on Automatic Control* 59.*4* (2014), pp. 1031–1036, DOI: 10.1109/TAC.2013.2285779.

[24] Pontus Giselsson et al., « Accelerated Gradient Methods and Dual Decomposition in Distributed Model Predictive Control », *in*: *Automatica* 49.*3* (2013), pp. 829–833, DOI: 10.1016/j.automatica.2013.01.009.

[25] Ronald Glowinski and Patrick Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, vol. 9, SIAM, 1989.

[26] Arun Gupta, Sharad Bhartiya, and P. S. V. Nataraj, « A Novel Approach to Multiparametric Quadratic Programming », en, *in*: *Automatica* 47.*9* (Sept. 2011), pp. 2112–2117, ISSN: 0005-1098, DOI: 10.1016/j.automatica.2011.06.019.

[27] Florian Herzog, *Strategic Portfolio Management for Long-Term Investments: An Optimal Control Approach*, ETH Zurich, 2005.

[28] T. A. Johansen and A. Grancharova, « Approximate Explicit Constrained Linear Model Predictive Control via Orthogonal Search Tree », *in*: *IEEE Transactions on Automatic Control* 48.*5* (May 2003), pp. 810–815, ISSN: 1558-2523, DOI: 10.1109/TAC.2003.811259.

[29] Norman H. Josephy, *Newton's Method for Generalized Equations.* Tech. rep., Wisconsin Univ-Madison Mathematics Research Center, 1979.

[30] Jay H Lee, « Model Predictive Control: Review of the Three Decades of Development », *in*: *International Journal of Control, Automation and Systems* 9.*3* (2011), p. 415, DOI: 10.1007/s12555-011-0300-6.

[31] Johan Löfberg, « YALMIP: A Toolbox for Modeling and Optimization in MATLAB », *in*: *Proceedings of the CACSD Conference*, vol. 3, Taipei, Taiwan, 2004, DOI: 10.1109/CACSD.2004.1393890.

[32] Jan R. Magnus and Heinz Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons, 2019.

[33] David Q. Mayne, « Model Predictive Control: Recent Developments and Future Promise », *in*: *Automatica* 50.*12* (2014), pp. 2967–2986, DOI: 10.1016/j.automatica.2014.10.128.

[34] David Q Mayne et al., « Constrained Model Predictive Control: Stability and Optimality », *in*: *Automatica* 36.*6* (2000), pp. 789–814, DOI: 10.1016/S0005-1098(99)00214-9.

[35] Sanjay Mehrotra, « On the Implementation of a Primal-Dual Interior Point Method », *in*: *SIAM Journal on optimization* 2.*4* (1992), pp. 575–601, DOI: 10.1137/0802028.

[36] Ruth Mitze and Martin Mönnigmann, « A Dynamic Programming Approach to Solving Constrained Linear–Quadratic Optimal Control Problems », en, *in*: *Automatica* 120 (Oct. 2020), p. 109132, ISSN: 0005-1098, DOI: 10.1016/j.automatica.2020.109132.

[37] Martin Mönnigmann, « On the Structure of the Set of Active Sets in Constrained Linear Quadratic Regulation », en, *in*: *Automatica* 106 (Aug. 2019), pp. 61–69, ISSN: 0005-1098, DOI: 10.1016/j.automatica.2019.04.017.

[38] João FC Mota et al., « Distributed ADMM for Model Predictive Control and Congestion Control », *in*: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 5110–5115, DOI: 10.1109/CDC.2012.6426141.

[39] Angelia Nedić, Alex Olshevsky, and Wei Shi, « Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs », *in*: *SIAM Journal on Optimization* 27.*4* (Jan. 2017), pp. 2597–2633, ISSN: 1052-6234, DOI: 10.1137/16M1084316.

[40] Angelia Nedic and Asuman Ozdaglar, « Distributed Subgradient Methods for Multi-Agent Optimization », *in*: *IEEE Transactions on Automatic Control* 54.*1* (Jan. 2009), pp. 48–61, ISSN: 1558-2523, DOI: 10.1109/TAC.2008.2009515.

[41] Yurii Nesterov, *Lectures on Convex Optimization*, vol. 137, Springer, 2018.

[42] Yurii E Nesterov, « A Method for Solving the Convex Programming Problem with Convergence Rate O (1/K$\hat{2}$) », *in*: *Dokl. Akad. Nauk Sssr*, vol. 269, 1983, pp. 543–547.

[43] JORGE NOCEDAL and Stephen J Wright, *NUMERICAL OPTIMIZATION*. Springer, 2006.

[44] Christina Oberlin and Stephen J. Wright, « Active Set Identification in Nonlinear Programming », *in*: *SIAM Journal on Optimization* 17.*2* (Jan. 2006), pp. 577–605, ISSN: 1052-6234, DOI: 10.1137/050626776.

[45] Panagiotis Patrinos and Alberto Bemporad, « An Accelerated Dual Gradient-Projection Algorithm for Embedded Linear Model Predictive Control », *in*: *IEEE Transactions on Automatic Control* 59.*1* (2014), pp. 18–33, DOI: 10.1109/TAC.2013.2275667.

[46] Panagiotis Patrinos and Haralambos Sarimveis, « A New Algorithm for Solving Convex Parametric Quadratic Programs Based on Graphical Derivatives of Solution Mappings », en, *in*: *Automatica* 46.*9* (Sept. 2010), pp. 1405–1418, ISSN: 0005-1098, DOI: 10.1016/j.automatica.2010.06.008.

[47] Warren B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 703, John Wiley & Sons, 2007.

[48] James A. Primbs, « Dynamic Hedging of Basket Options under Proportional Transaction Costs Using Receding Horizon Control », *in*: *International Journal of Control* 82.*10* (2009), pp. 1841–1855, DOI: 10.1080/00207170902783341.

[49] S. Joe Qin and Thomas A. Badgwell, « An Overview of Industrial Model Predictive Control Technology », *in*: *AIche Symposium Series*, vol. 93, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997, pp. 232–256.

[50] Christopher V. Rao, Stephen J. Wright, and James B. Rawlings, « Application of Interior-Point Methods to Model Predictive Control », *in*: *Journal of optimization theory and applications* 99.*3* (1998), pp. 723–757, DOI: 10.1023/A:1021711402723.

[51] James B. Rawlings and Kenneth R. Muske, « The Stability of Constrained Receding Horizon Control », *in*: *IEEE transactions on automatic control* 38.*10* (1993), pp. 1512–1516, DOI: 10.1109/9.241565.

[52] J. Richalet et al., « Model Algorithmic Control of Industrial Processes », *in*: *IFAC Proceedings Volumes* 10.*16* (1977), pp. 103–120, DOI: 10.1016/S1474-6670(17)69513-2.

[53] Jacques Richalet et al., « Model Predictive Heuristic Control », *in*: *Automatica (journal of IFAC)* 14.*5* (1978), pp. 413–428, DOI: 10.1016/0005-1098(78)90001-8.

[54] J. A. Rossiter and B. Kouvaritakis, « Constrained Stable Generalised Predictive Control », *in*: *IEE Proceedings D-Control Theory and Applications*, vol. 140, IET, 1993, pp. 243–254.

[55] Pierre OM Scokaert and James B. Rawlings, « Infinite Horizon Linear Quadratic Control with Constraints », *in*: *IFAC Proceedings Volumes* 29.*1* (1996), pp. 5905–5910, DOI: 10.1016/S1474-6670(17)58626-7.

[56] Maria M. Seron, Jose A. De Dona, and Graham C. Goodwin, « Global Analytical Model Predictive Control with Input Constraints », *in*: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 1, IEEE, 2000, pp. 154–159, DOI: 10.1109/CDC.2000.912749.

[57] Jonathan Richard Shewchuk, *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, Carnegie-Mellon University. Department of Computer Science, 1994.

[58] Wei Shi et al., « Extra: An Exact First-Order Algorithm for Decentralized Consensus Optimization », *in*: *SIAM Journal on Optimization* 25.*2* (2015), pp. 944–966.

[59] Kalyan T. Talluri and Garrett J. Van Ryzin, *The Theory and Practice of Revenue Management*, vol. 68, Springer Science & Business Media, 2006.

[60] Petter Tondel, Tor Arne Johansen, and Alberto Bemporad, « Further Results on Multiparametric Quadratic Programming », *in*: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 3, IEEE, 2003, pp. 3173–3178, DOI: 10.1109/CDC.2003.1273111.

[61] Petter TøNdel, Tor Arne Johansen, and Alberto Bemporad, « An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions », *in*: *Automatica* 39.*3* (2003), pp. 489–497.

[62] Bo Wahlberg et al., « An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems », *in*: *IFAC Proceedings Volumes* 45.*16* (2012), pp. 83–88, DOI: 10.3182/20120711-3-BE-2027.00310.

[63] Yang Wang and Stephen Boyd, « Fast Model Predictive Control Using Online Optimization », *in*: *IEEE Transactions on control systems technology* 18.*2* (2009), pp. 267–278, DOI: 10.1109/TCST.2009.2017934.

[64] Robert WM Wedderburn, « Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss—Newton Method », *in*: *Biometrika* 61.*3* (1974), pp. 439–447.

[65] Elizabeth Lai Sum Wong, « Active-Set Methods for Quadratic Programming », en, PhD thesis, UC San Diego, 2011.

[66] Stephen J. Wright, « Interior Point Methods for Optimal Control of Discrete Time Systems », *in*: *Journal of Optimization Theory and Applications* 77.*1* (1993), pp. 161–187, DOI: 10.1007/BF00940784.

[67] Stephen J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[68] Hiroshi Yamashita and Hiroshi Yabe, « Superlinear and Quadratic Convergence of Some Primal-Dual Interior Point Methods for Constrained Optimization », *in*: *Mathematical Programming* 75.*3* (1996), pp. 377–397.

[69] Tao Yang et al., « A Survey of Distributed Optimization », *in*: *Annual Reviews in Control* (2019), DOI: 10.1016/j.arcontrol.2019.05.006.

[70] A. Zhang and Manfred Morari, « Stability of Model Predictive Control with Soft Constraints », *in*: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 2, IEEE, 1994, pp. 1018–1023, DOI: 10.1109/CDC.1994.411277.

[71] Kunwu Zhang and Yang Shi, « Adaptive Model Predictive Control for a Class of Constrained Linear Systems with Parametric Uncertainties », *in*: *Automatica* 117 (2020), p. 108974, DOI: 10.1016/j.automatica.2020.108974.

[72] A. Zheng and M. Morari, « Stability of Model Predictive Control with Mixed Constraints », *in*: *IEEE Transactions on Automatic Control* 40.*10* (Oct. 1995), pp. 1818–1823, ISSN: 1558-2523, DOI: 10.1109/9.467664.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

CentraleSupélec

**Titre :** La terminaison Accélérée dans les Itérations Basées sur la Décomposition Duale pour la Commande Prédictive

**Mot clés :** Commande prédictive, Sous-optimalité, Optimisation convexe, Décomposition duale, Faisabilité.

**Résumé :** La commande prédictive (MPC) a suscité un intérêt croissant au cours des dernières décennies pour sa capacité à livrer des actions de commande optimales tout en satisfaisant les contraintes. Cependant, la solution optimale du problème d'optimisation résultant est parfois difficile à obtenir en pratique en raison de l'exigence d'échantillonnage rapide ou des limites de puissance de calcul. La décomposition duale, qui permet d'intégrer les contraintes et les interactions du système, est depuis longtemps une façon attrayante de traiter le problème. Un processus itératif est mis en œuvre pour déterminer la solution souhaitée. Bien que convergeant vers la solution optimale, l'optimalité et la faisabilité ne sont garanties que dans la limite des itérations. Dans cette thèse, de nouvelles conditions d'arrêt, avec garantie de sous-optimalité et de faisabilité, sont proposées pour obtenir des solutions sous-optimales et accélérer la fin du processus itératif. Cette idée d'une terminaison accélérée est explorée dans diverses configurations utilisant différentes méthodes itératives, pour lesquelles les preuves théoriques correspondantes sont fournies, et l'efficacité est illustrée par des exemples numériques. Le travail proposé, y compris les conditions d'arrêt et les algorithmes pour résoudre les solutions sous-optimales, peut être appliqué soit aux problèmes résultants de MPC avec des formulations spécifiques, soit à l'optimisation convexe générale.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

CentraleSupélec

**Title:** Accelerated Termination in Dual Decomposition Based Iterations for Model Predictive Control

**Keywords:** Model predictive control, Suboptimality, Convex optimization, Dual decomposition, Feasibility

**Abstract:** Model Predictive Control (MPC) has attracted increasing interest over the last decades for its capability in delivering optimal control actions while satisfying constraints. However, the optimal solution of the resulting optimization problem is sometimes intractable to acquire in practice due to rapid sampling requirements or computing power limits. Dual decomposition, competent in integrating constraints and interactions of the system, has long been an appealing way to treat the problem. Subsequently, an accessible iterative process proceeds to determine the desired solution. Although converging towards the optimal solution, the optimality and feasibility are only guaranteed in the limit of iterations. In this dissertation, new stopping conditions, with suboptimality and feasibility guarantee, are proposed to obtain suboptimal solutions and accelerate the termination of the iterative process. The idea of accelerated termination is explored in various configurations using different iterative methods, in which the corresponding theoretical proofs are provided, and the effectiveness is illustrated through numerical examples. The proposed work, including stopping conditions and algorithms to solve suboptimal solutions, can be applied either to MPC resulting problems with specific formulations or to general convex optimization.