Efficient algorithms for embedded optimisation-based control

Dang, Van Thuy

2018

Dang, V. T. (2018). Efficient algorithms for embedded optimisation-based control. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/89940>

<https://doi.org/10.32657/10220/46450>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

Efficient Algorithms for Embedded Optimisation-based Control

DANG VAN THUY

**Interdisciplinary Graduate School
Energy Research Institute @ NTU**

2018

Efficient Algorithms for Embedded Optimisation-based Control

DANG VAN THUY

(Supervised by: Prof. Keck Voon Ling)

INTERDISCIPLINARY GRADUATE SCHOOL

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

Acknowledgements

I feel indebted to my supervisor, Prof. Keck Voon Ling, for his insightful guidance, enormous patience and continuous support during my PhD journey. I have gained valuable problem-thinking and problem-solving skills under his supervising.

I would especially like to thank Prof. Jan Maciejowski for hosting me at the Control Group at University of Cambridge in July 2015, and for giving me a lot of valuable suggestions so that I could gain progress in research.

I also would like to thank Prof. Daniel E. Quevedo (University of Paderborn, Germany) for his support in starting and finishing some research projects.

Within Nanyang Technological University, I would like to thank my co-supervisor Prof. Douglas Leslie Maskell and my mentor Prof. Soh Yeng Chai for their time and necessary support. In addition, I would like to thank Dr. Cao-Tri Tran for useful talks during lunch times.

Outside Nanyang Technological University, I would like to thank Michael Baranski and Mihai Huzmezan from A. Eberle GmbH for providing me the industrial device and development tools.

Last but definitely not least, without endless support from my parents and my wife, I could never achieve this goal.

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Date

Signature

Authorship Attribution Statement

This thesis contains material from papers that were published/submitted in the following conferences/journals where I was the first author or co-author.

Chapter 3 is based on “T. V. Dang, K. V. Ling and J. M. Maciejowski. Banded null basis and ADMM for embedded MPC. *The 20th World Congress of the International Federation of Automatic Control, Toulouse-France, 2017, pp. 13712-13717.*”

Chapter 4 is based on “T. V. Dang, K. V. Ling and J. M. Maciejowski. Embedded ADMM-based QP solver for MPC with polytopic constraints. *European Control Conference (ECC), Linz-Austria, 2015, pp. 3446-3451.*”

The contributions of the co-authors in above papers are as follows:

- T. V. Dang (author of this thesis) initiated the idea, prepared the draft. Prof. K.V. Ling (main supervisor) and Prof. J.M. Maciejowski (University of Cambridge) suggested changes, polished the words and sentences.

Section 5.1, 5.2, 5.5.1, 5.5.2 in Chapter 5 are based on “K. Q. Huang, T. V. Dang, K. V. Ling and D. E. Quevedo. Event-triggered anytime control with two controllers. *54th IEEE Conference on Decision and Control (CDC), Osaka-Japan, 2015, pp. 4157-4162.*”

The contributions of the co-authors in sections are as follows:

- Prof. K.V. Ling initiated the idea to extend Prof. Quevedo’s work and collaborate with him.
- K.Q. Huang prepared the initial draft. T.V. Dang enhanced the paper in terms of mathematical aspects.
- Prof. K.V. Ling and T.V. Dang solved the combinatoric problem arising from the State-dependent Random-time Drift condition when there are two control laws .
- Prof. Quevedo (University of Paderborn, Germany) suggested changes, polished the language.

Section 5.3, 5.4 and 5.5.3 in Chapter 5 are based on “T. V. Dang, K.V. Ling and D.E. Quevedo. Stability analysis of event-triggered anytime control with multiple control laws. *IEEE Transaction on Automatic Control, 2018.*”

The contributions of the co-authors in these sections are as follows:

- Prof. K.V. Ling and Prof. Quevedo initiated the preliminary idea.

- T.V. Dang finalized the idea and prepared the draft.
- Prof. Quevedo and Prof. K.V. Ling suggested changes, polished the language.

Abstract

Solving optimisation problem is computationally demanding, and hence Model Predictive Control (MPC), an optimisation-based control technology, is traditionally employed in applications with slow dynamics. In recent years, due to its ability to systematically handling constraints and multiple-input and multiple-output systems, MPC has been extended to many non-traditional areas, in particular networked or embedded applications. However, limited computational resources poses challenges for embedded implementation of MPC. Computational resources to solve MPC problem may be time-varying and insufficient at times. Moreover, the measurements transmitted through a communication network may be unavailable due to network congestion or packet dropouts. To enable MPC be more widely used in networked or embedded applications, this thesis addresses two aspects: (1) methods to efficiently solve the optimisation problem in embedded platforms and (2) methods to tackle time-varying computational resources.

System of linear equations with the saddle point type is the main computational load in some MPC-tailored algorithms. Existing MPC-tailored algorithms have not exploited banded null bases in the sparse MPC formulation. In this thesis, an algorithm for exploiting banded null bases is presented. The proposed algorithm is tested with a wide range of MPC benchmark problems. Implementation results on an industrial embedded platform confirm that significant reduction in the runtime for the Alternating Direction Method of Multipliers (ADMM) can be achieved by the proposed algorithm. Methods for improving the conditioning of the banded bases are also developed.

Event-triggered Sequence-based Anytime Control (E-SAC), recently proposed in the literature, can handle the time-varying computational resources effectively. The main idea of E-SAC is, when computing resources and measurements are available, to compute a sequence of tentative control inputs and store them in a buffer for potential future use. Existing E-SAC in the literature only features one control law in the buffer. In this thesis, E-SAC is extended to schemes featuring multiple control laws. Numerical simulations show that performance in terms of empirical closed-loop cost, channel utilisation and regions for stochastic stability guarantees could be improved, compared with the existing E-SAC. However, the current stability analysis framework for E-SAC, the State-dependent Random-time Drift approach, becomes combinatoric and difficult to use when extending to the multi-control law E-SAC. To overcome this, a new stability analysis method for E-SAC based on Markov jump system is developed. Using the proposed stability analysis method, stochastic stability conditions of existing E-SAC are also recovered as a special case. In addition, the proposed technique systematically extends to other more sophisticated E-SAC scheme for which, until now, no analytical expression had been obtained.

Lay summary

Feedback control plays an essential role in human activities. Operations of modern machines such as airplanes, cars, robots, ships, all involves feedback control so that human can operate them with ease. The quality of the feedback control system is determined by the control method. Among various feedback control methods, Model Predictive Control (MPC) has shown various strengths and demonstrated outstanding advantages in the petro-chemical process industry - its traditional field. However, there are challenges when extending MPC to some non-traditional fields, e.g. networked or embedded systems with a small computing chip. This thesis proposes methods which will enable the implementation of MPC in networked or embedded systems. As a result, the advantages of MPC can be further extended to a wider range of applications.

Contents

List of Figures	xiv
List of Tables	xv
Abbreviations	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Summary of contributions	2
1.3 Thesis overview	4
2 Related backgrounds	5
2.1 Model Predictive Control	5
2.2 Optimisation algorithms for embedded Model Predictive Control	8
2.3 Saddle point type problem and the null space method	13
2.4 Event-triggered sequence-based anytime control	16
3 Banded null basis and ADMM for embedded MPC	18
3.1 Banded null basis to solve KKT system	18
3.2 Banded reduced-Hessian QP problem	20
3.3 Schur complement reduction for KKT system	22
3.4 On the computation and the existence of a banded null basis in MPC . . .	24

3.4.1	Turnback algorithm	24
3.4.2	New algorithm for efficiently computing the banded null basis, and the existence of banded null basis in MPC	25
3.5	Improving the condition number of the banded null basis	30
3.6	Case study 1: a set of MPC problems tested in an industrial device	33
3.7	Case study 2: embedded quadrotor application	39
3.8	Automatic code generation for embedded implementation of MPC	42
3.9	Conclusion	43
3.10	Appendix	43
4	An ADMM algorithm for handling polytopic constraints and its fixed- point arithmetic implementation on a FPGA	45
4.1	Algorithm description	46
4.2	Step-size selection methods	47
4.2.1	Matrix recurrence form	47
4.2.2	Step-size selection	49
4.3	Implementation on an FPGA	52
4.3.1	Offline and online computations	52
4.3.2	Fixed-point arithmetic implementation	53
4.4	Case study	53
4.4.1	Illustration example	53
4.4.2	Result and discussion	54
4.4.3	Comparison between 2-norm and max-norm	55
4.5	Conclusion	56
5	Event-triggered anytime control	57
5.1	System model	58
5.2	E-SAC with multiple control laws	61

5.2.1	Algorithm <i>B2</i> : two-control law E-SAC without buffer	62
5.2.2	Algorithm <i>A2</i> : two-control law E-SAC with buffer	62
5.2.3	State-dependent random-time drift condition approach for stability analysis of E-SAC	64
5.3	Markov Jump System model and stability framework	64
5.3.1	Markov state of the buffer content	65
5.3.2	Markov jump system with event triggering model	68
5.3.3	General model and stability analysis framework	69
5.3.4	Stochastic stability of E-MJS model	70
5.4	Stochastic Stability of E-SAC Schemes	72
5.4.1	Existence and bounds of $\tilde{V}(y_k)$ of E-SAC	72
5.4.2	Stochastic stability for E-SAC schemes <i>A1</i> and <i>A2</i>	72
5.4.3	Recover stability of <i>A1</i>	73
5.5	Numerical simulation	74
5.5.1	Performance	75
5.5.2	Stability regions	75
5.5.3	Stability region with respect to the computational cost of fine policy	77
5.6	Conclusion	79
5.7	Appendix	80
6	Conclusions and future works	89
6.1	Conclusion	89
6.2	Future work	90
	Bibliography	93
	Publications	100

List of Figures

2.1	Principle of MPC	6
2.2	Patterns of matrices in the sparse QP formulation.	8
2.3	Patterns of matrices in the condensed QP formulation.	9
2.4	Control system with the event-triggered sequence-based anytime algorithm (E-SAC)	16
3.1	Pattern of $Y^T(H + \rho G^T G)Y$ in (3.1.2) (center sub-figure) and its Cholesky factorisation matrix (right figure, the "lower" form of Cholesky factorisation is used) (MPC problem with 6 states & 2 inputs & control horizon 10) . . .	20
3.2	Matrices in the banded reduced-Hessian QP	21
3.3	Pattern of $F\tilde{H}^{-1}F^T$ in (3.3.13) (left figure) and its Cholesky factorisation matrix (right figure, the "lower" form of Cholesky factorisation is used) (MPC problem with 6 states & 2 inputs & control horizon 10)	23
3.4	The diagonal line of the banded basis passing through the left-top and right-bottom corners; point P : last non-zero element of column Y_2 ; red arrow: the Euclidean distance from P to the diagonal line.	28
3.5	Banded null basis Y for the null space of matrix F arising in MPC problem with plant model of [1] [2].	32
3.6	Hardware used for the implementation: EOR-3D (ARM-9, CPU 400MHz, RAM 64MB, embedded Linux)	33
3.7	Sub-plot 1 and 2: ratio of time-per-iteration of banded null basis method (B.b) versus two other methods (standard and Schur reduction). Sub-plot 3: Ratio of time-per-iteration of reduced-Hessian QP using banded null basis versus using condensed formulation. Vertical axis: ratio $\frac{n_x}{n_u}$	35

3.8	Convergence in terms of runtime (of direct ADMM) for MPC problem 5 (Double inverted pendulum). Banded null basis method (with condition tuning) in double precision approaches the “fine” sub-optimal solution fastest.	37
3.9	Banded null basis in 18 MPC problems ($N = 10$).	38
3.10	Camera field-of-view quadrotor system [3]	39
3.11	Sparsity pattern of F and its null space basis Y in the camera-field-of-view quadrotor problem.	41
3.12	Time per ADMM iteration versus prediction horizon of reduced-Hessian-based ADMM using banded null basis	42
3.13	MPC controller based on the proposed ADMM algorithms for a tap-changer transformer control problem	43
4.1	Basic FPGA architect consisting of configurable blocks	52
4.2	Spring-mass System	53
4.3	MPC performance in Spring-Mass system	55
4.4	Selection of ρ based on the 2-norm or max-norm criteria. (MPC parameters: $Q_x = C_d^T C_d$, $R = I$)	55
4.5	Selection of ρ based on the 2-norm or max-norm criteria. (Random MPC parameters: $Q_x \succeq 0$, $R \succ 0$)	56
5.1	Setup of a system with an Event-triggered Sequence-based Anytime (E-SAC) algorithm.	58
5.2	Operation of $A1$, b_k : buffer content at time instant k .	61
5.3	Operation of $A2$, $b_k^\dagger$: buffer content at time instant k .	63
5.4	Illustration of the State-dependent Random-time Drift condition approach in the E-SAC with two control laws. $\{k_0, k_1, \dots\}$: time steps that the buffer is empty, $\Delta_i = k_{i+1} - k_i$ and r_i (number of times the coarse control law is active during $[k_i, k_{i+1}]$) are random variables.	65
5.5	State transition diagram for buffer length of $A1$ ($N_{max} = 4$), when $ x_k > d$	66
5.6	State transition diagram for buffer contents of $A2$ ($N_{max} = 4$, $\eta = 2$), $ x_k > d$. During periods when $ x_k \leq d$, the buffer is always emptied, i.e. there is no change in the buffer content.	67

5.7	E-MJS model of E-SAC schemes.	69
5.8	Performance, in term of a trade-off curve between channel utilisation CU and empirical cost J	76
5.9	Boundaries of stability: algorithms $B1$ (red, circle) and $B2$ (blue, with $\epsilon = 1$ (dashed), $\epsilon = 0.8$ (dotted), $\epsilon = 0.5$ (dash-dot), $\epsilon = 0.05$ (solid)).	76
5.10	Boundaries of stability: $A1$ (red, square) and $A2$ (blue with $\epsilon = 1$ (dashed), $\epsilon = 0.8$ (dotted), $\epsilon = 0.5$ (dash-dot), $\epsilon = 0.05$ (solid)).	78
5.11	Stability guarantee region for $A2$ with respect to η (processing units needed to compute κ_2) and $\epsilon = \frac{\rho_2}{\rho_1}$. Two-control scheme $A2$ with configuration $Q1$ guarantees to yield a stable system if the open-loop bound satisfying $\alpha <$ 1.3527 . Two-control scheme $A2$ with configuration $Q2$ guarantees to yield a stable system if $\alpha < 1.266$. One-control scheme $A1$ with configuration $Q3$ guarantees to yield a stable system if $\alpha < 1.175$. System (5.5.25) has open-loop bound $\alpha = 1.35$	79
5.12	Averaged value of $V(x) = x_k $ over 10^4 random realizations.	80

List of Tables

3.1	Time (ms) per iteration of ADMM with different implementation methods	36
3.2	Conditioning of banded null basis obtained using different methods (M2+M1: Apply Method 2 then apply Method 1. Similarly for M2A+M1.)	38
3.3	Percentage increase of non-zero entries when trading sparsity for conditioning	39
3.4	<i>Ave. runtime (ms)/ ave. iteration</i> for camera-field-of-view problem. The runtime is averaged over 100 random initial states. Notation: ADMM-B-RH : ADMM with the reduced Hessian and banded null basis; ##: CVXGEN (an interior-point QP algorithm of [4]) is unable to generate C-code for big problem.	40
3.5	Code size (kB) (Notation: ADMM-B-RH : ADMM with the reduced Hessian and banded null basis; ##: CVXGEN unable to generate code for big dimension)	40

Abbreviations

ADMM	Alternating Direction Method of Multipliers
ASM	Active-set method
E-SAC	Event-triggered sequence-based anytime control
FPGA	Field Programmable Gate Array
IPM	Interior-point method
KKT	Karush-Kuhn-Tucker
MPC	Model Predictive Control
QP	Quadratic programming
SAC	Sequence-based anytime control
SRD	State-dependent Random-time Drift

Chapter 1

Introduction

1.1 Motivation

Since its appearance in the 1970s [5] [6], Model Predictive Control (MPC) has had an exceptional history [7] due to its ability in handling multiple inputs and outputs with constraints. Traditionally, MPC has been popularly employed in process industry where the system dynamics is slow, hence there is sufficient time for solving the computationally demanding optimisation problems. In recent years, model predictive control for small but fast dynamic embedded systems has been receiving increased attention [7]. In these embedded systems, the computing platforms, e.g. a microcontroller, usually have limited computation resources. In addition, the real-time requirement, especially in safety-critical applications, cannot be ignored. Existing iterative-based optimisation algorithms for MPC often need to solve a system of linear equations of a saddle point type in each iteration, see e.g. interior-point based algorithm in [8]. The saddle point type problem is usually referred to as KKT (Karush-Kuhn-Tucker) system in the optimisation literature [9]. In most of cases, it is the main computational load. Therefore, accelerating the solution of the KKT system will substantially improve the performance of many MPC optimisation algorithms.

It is common in the embedded or networked control system that processor availability varies due to varying computational loads and multi-tasking operations. In addition, when sensor measurements are transmitted through a communication network, the measurements may be unavailable due to packet dropouts, or network congestion. Event-triggered sequence-based anytime control (E-SAC) [10] [11] could deal with these situation effectively. The main idea of E-SAC is, when computing resources and measurements are available, to compute a sequence of tentative control inputs and store them in a buffer for potential future use. Currently, existing E-SAC only features one control law in the

buffer. In control system design, one may encounter situations where one would like to switch between different control laws in respond to changing operating conditions. For example, one may switch between a suboptimal or optimal solution in MPC [12], a fixed- or floating-point controller implementation [13]. However, E-SAC with two or more control laws with different computational cost has not been considered in the literature.

1.2 Summary of contributions

Contributions to MPC-tailored optimisation algorithms

Methods for solving a system of linear equations include two main families: (1) direct methods and (2) iterative methods. Direct methods, such as Gaussian elimination and QR decomposition, find the exact solution after a finite number of steps. They are more robust, reliable, predictable than iterative methods [14]. However, they require a high computation cost. Iterative methods, such as Conjugate gradients and MINRES, solve the system of linear equations by repeatedly improving, from an initial guess, the solution until it is close enough to the exact solution. These methods usually require fewer operations and memory storage than direct methods but their efficiency relied on many problem-dependent parameters [14]. Existing MPC-tailored optimisation algorithms employs both direct and iterative methods to deal with the saddle-point-type system of linear equations. When formulating the optimisation problem of MPC as a sparse quadratic program, the matrices will have special structures such as bandedness and repeating pattern. These structures have been exploited in some interior-point methods, resulting in a significant reduction of the computational cost and a linear growth in computation time with the control horizon, see e.g. [8] [15] [16]. However, banded null bases have not exploited.

This thesis proposes a novel method exploiting a banded null basis to accelerate the computation of MPC for embedded applications. The banded basis is used to solve the saddle-point-type system of linear equations arising from several MPC-tailored optimisation algorithms. By using a banded null basis, the system of linear equations in each iteration is solved by solving smaller systems of linear equations with banded matrices. The banded null basis can also be used to obtain a reduced-Hessian QP problem with a banded Hessian. Banded matrices offer the following computational advantages: (1) smaller storage memory footprint and (2) fewer computing operations in a matrix-vector multiplication. These lead to improvement in computation as well as hardware implementation cost. The proposed null basis technique is incorporated within the Alternating Direction Method of Multipliers (ADMM) and demonstrates that the significant speedup can be achieved by exploiting the banded null basis.

In addition, a method for handling the polytopic constraints letting ADMM be more

applicable for a wide range of embedded MPC problems, and method for fixed-point arithmetic implementation leading to a cheaper hardware cost, e.g when using an FPGA, are also presented.

Contributions to event-triggered anytime control

MPC is a candidate for anytime algorithms. Indeed, for real-time applications of MPC, where processor availability is usually assumed fixed and known a priori, techniques such as early termination [17] or suboptimal MPC [12], which aim at providing suboptimal solutions to reduce the online computations and thereby meet real-time requirements could be adapted to a multi-control-law framework.

Existing E-SAC algorithm only features one control law in the buffer. Whereas, as aforementioned, in control system design there are situations where more than one control laws are desired. Furthermore, the State-dependent Random-time Drift (SRD) technique of [18] was the only available approach for analysing the stability of existing E-SAC. In the author's preliminary work [19], the SRD technique was used to analyse the stability of a multi-control law E-SAC scheme. For E-SAC with one control law, there is one random variable, and one can obtain an analytical expression for its probability mass function. For E-SAC with multiple control laws, there are more than one random variables describing all possibilities. As a result, obtaining an analytical expression seems to be combinatoric and intractable. It was concluded that extending SRD technique to more sophisticated E-SAC schemes will be difficult.

This thesis extends the E-SAC to schemes which feature multiple control laws. Novel stability analysis for the multi-control-law E-SAC schemes based on Markov jump system, which is more scalable than the existing in the literature - the State-dependent Random-time Drift condition, is developed.

Summary of major contributions

- Methods for exploiting banded null bases to accelerate the computation of saddle-point-type system of linear equations in some MPC-tailored algorithms. When incorporated within ADMM algorithms, it shows that the use of banded null basis reduces the runtime of ADMM significantly up to 26 times compared with a standard implementation.
- For a single-input system, the existence of a banded basis in MPC is proved, and an upper bound for its sparsity is derived. In case the banded basis is ill-conditioned, methods for trading its sparsity for conditioning are proposed.
- An ADMM-based algorithm that can handle polytopic constraints in MPC, and a fixed-point arithmetic implementation framework for this algorithm are developed.

- To extend the E-SAC scheme to handle multiple control laws, resulting in: (1) the communication and processing resources could be used more efficiently and (2) performance in terms of empirical closed-loop cost, channel utilisation and regions for stochastic stability guarantees could be improved, compared with the basic E-SAC.
- A new stability analysis framework based on Markov jump system is developed for the E-SAC with multiple control laws. This new framework is more scalable than the existing State-dependent Random-time Drift condition available in the literature. Using the new stability analysis, stochastic stability conditions of existing E-SAC are also recovered. In addition, the proposed technique systematically extends to a more sophisticated E-SAC scheme for which, until now, no analytical expression had been obtained.

1.3 Thesis overview

In Chapter 2, an overview of related background is provided.

Chapter 3 describes the proposed banded null basis methods to accelerate the solution of MPC. The advantage of the proposed method is demonstrated in two situations (1) solving the KKT system arising from ADMM and (2) transforming the sparse MPC formulation to a reduced-Hessian QP while maintaining the banded structure of the Hessian. To compute a banded null basis, turnback algorithm in the literature and a new proposed algorithm are presented. The existence of the banded basis in MPC is proved for single-input systems, and an upper bound for its sparsity density is derived. In case an ill-conditioned basis arises, methods for improving the conditioning are developed. The improved ADMM algorithms are implemented in an industrial device with a set of 15 MPC problems, and in the ARM-Cortex A9 processor with a quadrotor problem.

Chapter 4 describes an ADMM algorithm that can handle polytopic constraints and its fixed-point arithmetic implementation on an FPGA.

In Chapter 5, event-triggered anytime control with multiple control laws are proposed, and stability analysis framework based on Markov jump system is developed. The multi-control laws schemes outperforms the one-control scheme in a numerical simulation. In another numerical simulation, stochastic stability of the multi-control laws scheme is derived with different values of the involved parameters.

Chapter 6 draws the conclusion and proposes some future directions.

Chapter 2

Related backgrounds

This chapter provide an overview of existing technologies that are related to this thesis, including Model Predictive Control (MPC), saddle point type system in existing MPC-tailored algorithms and the null space method for solving it, the Alternating Direction Method of Multipliers (ADMM) and event-triggered anytime control (E-SAC).

2.1 Model Predictive Control

Fig. 2.1 shows the principle of MPC. At each sampling time instant, MPC controller determines a sequence of control inputs by minimising an objective function with constraints over a prediction horizon:

$$\min_{u_k} J(x_0) = \sum_{k=0}^{N-1} q(x_k, u_k) + p(x_N) \quad (2.1.1)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \quad (2.1.2)$$

$$x_k \in \mathbb{X}, \quad u_k \in \mathbb{U}, \quad x_N \in \mathbb{X}_f \quad (2.1.3)$$

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are the states and the control inputs at the sampling instant $k = 0, 1, 2, \dots$, respectively; x_0 is the current measurement; $x_{k+1} = f(x_k, u_k)$ represents the discrete-time model of the controlled system, $q(x_k, u_k)$ is the stage cost and $p(x_N)$ is the terminal cost; $N \in \mathbb{Z}_+$ is the prediction horizon; $\mathbb{X}$, $\mathbb{U}$, $\mathbb{X}_f$ denote the state constraint set, control constraint set, and terminal constraint set respectively.

Then only the first element of the control sequence is applied (see the oval in Fig. 2.1), and the control sequence is determined again in next sampling instant; this is the so-called receding horizon strategy. This thesis considers MPC with a linear system, quadratic stage

Figure 2.1: Principle of MPC

and terminal cost functions, and linear constraints in (2.1.1) as following

$$\min_u J(x_0) = \sum_{k=0}^{N-1} (x_k^T Q_x x_k + u_k^T R u_k) + x_N^T Q_N x_N \quad (2.1.4)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k \quad (2.1.5)$$

$$G_x x_k \leq g_x, \quad G_u u_k \leq g_u \quad (k = 0, 1, \dots, N-1),$$

$$G_N x_N \leq g_N$$

where x_0 is given (current measurement); N is the horizon; $Q_x, Q_N \in \mathbb{R}^{n_x \times n_x}$ are positive semi-definite matrices; $R \in \mathbb{R}^{n_u \times n_u}$ is positive definite matrix; $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ are the matrices of the discrete-time state space model; $G_x \in \mathbb{R}^{n_{ix} \times n_x}$, $G_N \in \mathbb{R}^{n_{it} \times n_x}$, $G_u \in \mathbb{R}^{n_{iu} \times n_u}$, $g_x \in \mathbb{R}^{n_{ix}}$, $g_N \in \mathbb{R}^{n_{it}}$, $g_u \in \mathbb{R}^{n_{iu}}$ are appropriate matrices and vectors describing the constraints of the system (n_{ix} , n_{it} , n_{iu} are the numbers of state constraints, terminal constraints and input constraints).

By using the sparse formulation which keeps both the states and control as decision variables

$$z = (u_0; x_1; u_1; x_2; \dots; u_{N-1}; x_N), \quad (2.1.6)$$

problem (2.1.4) can be written as a quadratic programming (QP) problems as follows:

$$\min_z \frac{1}{2} z^T H z \quad (2.1.7)$$

$$\text{s.t. } Fz = f \quad (2.1.8)$$

$$Gz \leq g \quad (2.1.9)$$

where $H \in \mathbb{R}^{n_d \times n_d}$ is positive semi-definite ($n_d = N(n_x + n_u)$ is the number of decision variables). $F \in \mathbb{R}^{n_{eq} \times n_d}$, $G \in \mathbb{R}^{n_{ineq} \times n_d}$, $q \in \mathbb{R}^{n_d}$, $f \in \mathbb{R}^{n_{eq}}$, $g \in \mathbb{R}^{n_{ineq}}$ (where $n_{eq} = Nn_x$, $n_{ineq} = (N-1)(n_{ix} + n_{iu}) + n_{it} + n_{iu}$ are total number of equality and inequality constraints, respectively); the matrices F, G, f, g, H have the following descriptions:

$$F = \begin{bmatrix} -B & I & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -A & -B & I & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & -A & -B & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -A & -B & I \end{bmatrix}, \quad (2.1.10)$$

$$f = \begin{bmatrix} Ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, g = \begin{bmatrix} \mathbf{1}_{N-1} \otimes [g_u; g_x] \\ g_u \\ g_N \end{bmatrix}, G = \begin{bmatrix} I_{N-1} \otimes \begin{bmatrix} G_u & 0 \\ 0 & G_x \end{bmatrix} & 0 & 0 \\ 0 & G_u & 0 \\ 0 & 0 & G_N \end{bmatrix},$$

$$H = \begin{bmatrix} I_{N-1} \otimes \begin{bmatrix} R & 0 \\ 0 & Q_x \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} R & 0 \\ 0 & Q_N \end{bmatrix} \end{bmatrix} \quad (2.1.11)$$

On the other hand, one can also formulate (2.1.7) as a condensed QP formulation [20] which only keeps the control as the decision variables. In details, the decision variables are chosen as $\theta = (u_0; u_1; \dots; u_{N-1})$. Then the MPC problem (2.1.7) becomes

$$\min_{\theta} \frac{1}{2} \theta^T H_c \theta + (x_0^T \Gamma_1) \theta \quad (2.1.12)$$

$$\text{s.t. } \Gamma_2 \theta \leq g_c + [\Gamma_3 x_0; 0_{Nn_u}] \triangleq \tilde{g}_c \quad (2.1.13)$$

where $H_c, \Gamma_1, \Gamma_2, \Gamma_3$ are briefly shown in the following with detailed steps can be found in [20].

$$Q_{blk} = \text{diag}(Q_x; Q_x; \dots; Q_x; Q_N), R_{blk} = \text{diag}(R; R; \dots; R), A_{blk} = [A; A^2; \dots; A^N], B_{blk} =$$

Figure 2.2: Patterns of matrices in the sparse QP formulation.

$$\begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots & 0 \\ A^{N-1}B & A^{N-2}B & \dots & \dots & B \end{bmatrix}, G_{xblk} = \text{diag}(G_x; G_x; \dots; G_x),$$

$$G_{ublk} = \text{diag}(G_u; G_u; \dots; G_u).$$

$$\begin{aligned} H_c &= B_{blk}^T \cdot Q_{blk} \cdot B_{blk} + R_{blk}, \quad \Gamma_1 = B_{blk}^T \cdot Q_{blk} \cdot A_{blk}, \\ \Gamma_2 &= [G_{xblk} \cdot B_{blk}; G_{ublk}], \quad \Gamma_3 = -G_{xblk} \cdot A_{blk} \end{aligned} \quad (2.1.14)$$

The sparse QP will have both equality and inequality constraints while the condensed QP has only inequality constraints. One main advantage of formulating a sparse QP is that the Hessian matrix has a banded structure and can be exploited for computational advantage [21], see Fig. 2.2 and 2.3 for the patterns of matrices in these formulations.

2.2 Optimisation algorithms for embedded Model Predictive Control

In some cases, e.g. small systems, the arising QP problem in MPC can be solved offline by multi-parametric programming [22]. In other cases, where online solution of this QP is needed, second order methods such as interior-point methods (IPM) [23] and active-set (ASM) [24] [25] are two commonly employed approaches. The main computational load of

Figure 2.3: Patterns of matrices in the condensed QP formulation.

IPM and ASM is the solution of a set of linear equations at every iteration, and this can be the bottleneck for embedded systems. IPM outperforms ASM in terms of scalability [26]. Field Programmable Gate Array (FPGA) an emerging embedded platform having a highly parallel computation capability could be used to tackle the computation demanding task of solving the system of linear equations, see [15, 27, 28]. Hybrid platforms consisting of an FPGA and a general-purpose processor are also becoming popular, and each computing unit has a specific strength that can be exploited to accelerate the computation of MPC, see e.g. [29–31]. Recently, first-order QP solvers, such as gradient-based methods [32–36] or the ‘alternating direction method of multipliers’ (ADMM) [35, 37, 38], have received significant interest because of their simpler structure, low cost implementation, efficient parallelism, and fixed-point arithmetic opportunity. Interest in ADMM for MPC can be found in [39–42].

Due to its simplicity in concept and underlying algebraic calculations, ADMM is gaining particularly popular for embedded MPC applications. Basically, the ADMM method solves the following optimisation problem

$$\min_{p, q} \quad \zeta(p) + \phi(q) \quad \text{s.t.} \quad \Psi p + \Upsilon q = c$$

where $p \in \mathbb{R}^h$, $q \in \mathbb{R}^l$; $\zeta(\cdot)$ and $\phi(\cdot)$ are convex functions; $\Psi \in \mathbb{R}^{m \times h}$, $\Upsilon \in \mathbb{R}^{m \times l}$, $c \in \mathbb{R}^m$.

Using an augmented Lagrangian defined as

$$L_\rho(p, q, \lambda) = \zeta(p) + \phi(q) + \lambda^T(\Psi p + \Upsilon q - c) + \frac{\rho}{2} \|\Psi p + \Upsilon q - c\|_2^2$$

the ADMM method consists of three main steps:

$$\begin{aligned} p_{k+1} &= \arg \min_p L_\rho(p, q_k, \tau_k) \\ q_{k+1} &= \arg \min_q L_\rho(p_{k+1}, q, \tau_k) \\ \tau_{k+1} &= \tau_k + (\Psi p_{k+1} + \Upsilon q_{k+1} - c) \end{aligned}$$

where $\tau = \frac{1}{\rho}\lambda$ is a scaled dual variable, and $\rho > 0$ is called as the step-size. Stopping criteria are based on the primal residual $r_k = \|\Psi p_k + \Upsilon q_k - c\|$ and the dual residual $d_k = \|\rho \Psi^T \Upsilon(q_{k+1} - q_k)\|$, see [35].

There are several ADMM algorithms for MPC problems. The differences in these variants are mainly in the way the MPC problem is formulated as the standard setup of ADMM. A list of setups can be found in [43]. Below are two ADMM algorithms for MPC that are well performing in terms of number of iterations needed.

Direct ADMM [44]

A copying variable y is introduced to QP problem (2.1.7), $y = z$, and the equivalent problem is

$$\min_{z, y} \quad \frac{1}{2} z^T H z + \mathcal{I}_{Fz=f}(z) + \mathcal{I}_{Gy \leq g}(y) \quad (2.2.15)$$

$$\text{s.t. } z - y = 0 \quad (2.2.16)$$

where $\mathcal{I}_{Fz=f}(z)$ and $\mathcal{I}_{Gy \leq g}(y)$ are the indicator functions ⁱ.

Applying the ADMM framework, the ADMM iteration for direct ADMM setup is as follows:

The augmented Lagrangian of the direct ADMM setup:

$$\begin{aligned} L(z, y, \lambda) &= \frac{1}{2} z^T H z + \mathcal{I}_{Fz=f}(z) + \mathcal{I}_{Gy \leq g}(y) + \lambda^T (z - y) + \frac{1}{2} \rho \|z - y\|^2 \\ &= \frac{1}{2} z^T (H + \rho I) z + (\lambda - \rho y)^T z + \mathcal{I}_{Fz=f}(z) + \frac{1}{2} \rho \|y\|^2 + \mathcal{I}_{Gy \leq g}(y) - \lambda^T y \end{aligned}$$

ⁱThe indicator function of a set $\mathcal{G}$, $\mathcal{I}_{\mathcal{G}}(\cdot)$, meaning that $\mathcal{I}_{\mathcal{G}}(e) = 0$ if $e \in \mathcal{G}$ and $\mathcal{I}_{\mathcal{G}}(e) = +\infty$ if $e \notin \mathcal{G}$.

The first step of direct ADMM:

$$\begin{aligned}
z_{k+1} &= \arg \min_z L(z, y_k, \lambda_k) \\
&= \arg \min_z \frac{1}{2} z^T (H + \rho I) z + (\lambda_k - \rho y_k)^T z + \mathcal{I}_{Fz=f}(z) + \frac{1}{2} \rho \|y_k\|^2 + \mathcal{I}_{Gy \leq g}(y_k) - \lambda_k^T y_k \\
&= \arg \min_z \frac{1}{2} z^T (H + \rho I) z + (\lambda_k - \rho y_k)^T z + \mathcal{I}_{Fz=f}(z)
\end{aligned}$$

equivalent to solving following equality-constrained optimisation

$$\begin{aligned}
&\min_z \frac{1}{2} z^T (H + \rho I) z + (\lambda_k - \rho y_k)^T z \\
&\text{subject to } Fz = f
\end{aligned} \tag{2.2.17}$$

The Lagrangian of optimisation problem (2.2.17) is

$$\bar{L}(z, \theta) = \frac{1}{2} z^T (H + \rho I) z + (\lambda_k - \rho y_k)^T z + \theta^T (Fz - f)$$

The optimality condition is

$$\partial \bar{L} / \partial z = 0 \tag{2.2.18}$$

$$\partial \bar{L} / \partial \theta = 0 \tag{2.2.19}$$

These lead to

$$(H + \rho I)z + (\lambda_k - \rho y_k) + F^T \theta = 0 \tag{2.2.20}$$

$$Fz - f = 0 \tag{2.2.21}$$

which is written as

$$\begin{bmatrix} H + \rho I & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} z \\ \theta \end{bmatrix} = \begin{bmatrix} -\lambda_k + \rho y_k \\ f \end{bmatrix} = \begin{bmatrix} -\rho \tau_k + \rho y_k \\ f \end{bmatrix} \tag{2.2.22}$$

The derivations of the second and third step are quite straight forward. In summary, the iteration steps of direct ADMM are

$$1. \begin{bmatrix} H + \rho I & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} z_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \rho(y_k - \tau_k) \\ f \end{bmatrix} \tag{2.2.23}$$

$$2. y_{k+1} = \mathbb{P}_{\mathcal{G}}(z_{k+1} + \tau_k) \tag{2.2.24}$$

$$3. \tau_{k+1} = \tau_k + z_{k+1} - y_{k+1} \tag{2.2.25}$$

where τ_k, θ_k are Lagrangian multipliers, $\mathcal{G} = \{y | Gy \leq g\}$, and the projection operator

$$\mathbb{P}_{\mathcal{G}}(\bar{v}) = \arg \min_{v \in \mathcal{G}} \|v - \bar{v}\|$$

The second step, the projection (2.2.24), is reduced to clipping operation which is simple if the set $\mathcal{G}$ is a box-type constraint. Whereas if the set $\mathcal{G}$ is a polytopes, the projection will be computationally intensive hence this ADMM setup is not recommended in this general case.

Indirect ADMM [43]

In this setup, the slack variables s are introduced to the inequality constraints, i.e. $Gz + s = g$ with $s \geq 0$. The QP formulation is then:

$$\min_{z, y} \quad \frac{1}{2} z^T H z + \mathcal{I}_{Fz=f}(z) + \mathcal{I}_{s \geq 0}(s) \quad (2.2.26)$$

$$\text{s.t.} \quad Gz + s = g \quad (2.2.27)$$

The augmented Lagrangian of the indirect ADMM setup:

$$\begin{aligned} L(z, s, \lambda) &= \frac{1}{2} z^T H z + \mathcal{I}_{Fz=f}(z) + \mathcal{I}_{s \geq 0}(s) + \lambda^T (Gz + s - g) + \frac{1}{2} \rho \|Gz + s - g\|^2 \\ &= \frac{1}{2} z^T (H + \rho G^T G) z + (G^T \lambda + \rho G^T (s - g))^T z + \mathcal{I}_{Fz=f}(z) \\ &\quad + \frac{1}{2} \rho \|s\|^2 + \mathcal{I}_{s \geq 0}(s) - (g + \lambda)^T s + g^T g \end{aligned}$$

The first step of indirect ADMM:

$$\begin{aligned} z_{k+1} &= \arg \min_z L(z, s_k, \lambda_k) \\ &= \arg \min_z \frac{1}{2} z^T H z + \mathcal{I}_{Fz=f}(z) + \mathcal{I}_{s \geq 0}(s) + \lambda_k^T (Gz + s_k - g) + \frac{1}{2} \rho \|Gz + s_k - g\|^2 \\ &= \frac{1}{2} z^T (H + \rho G^T G) z + (G^T \lambda_k + \rho G^T (s_k - g))^T z + \mathcal{I}_{Fz=f}(z) \\ &\quad + \frac{1}{2} \rho \|s_k\|^2 + \mathcal{I}_{s \geq 0}(s_k) - (g + \lambda_k)^T s + g^T g \\ &= \arg \min_z \frac{1}{2} z^T (H + \rho G^T G) z + (G^T \lambda_k + \rho G^T (s_k - g))^T z + \mathcal{I}_{Fz=f}(z) \end{aligned}$$

equivalent to solving following equality-constrained optimisation

$$\begin{aligned} \min_z \quad & \frac{1}{2} z^T (H + \rho G^T G) z + (G^T \lambda_k + \rho G^T (s_k - g))^T z \\ \text{subject to} \quad & Fz = f \end{aligned} \quad (2.2.28)$$

The Lagrangian of optimisation problem (2.2.17) is

$$\bar{L}(z, \theta) = \frac{1}{2}z^T(H + \rho G^T G)z + (G^T \lambda_k + \rho G^T(s_k - g))^T z + \theta^T(Fz - f)$$

The optimality condition is

$$\partial \bar{L} / \partial z = 0 \quad (2.2.29)$$

$$\partial \bar{L} / \partial \theta = 0 \quad (2.2.30)$$

These lead to

$$(H + \rho G^T G)z + (G^T \lambda_k + \rho G^T(s_k - g)) + F^T \theta = 0 \quad (2.2.31)$$

$$Fz - f = 0 \quad (2.2.32)$$

which is written as

$$\begin{bmatrix} H + \rho G^T G & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} z \\ \theta \end{bmatrix} = \begin{bmatrix} -(G^T \lambda_k + \rho G^T(s_k - g)) \\ f \end{bmatrix} = \begin{bmatrix} -\rho G^T(\tau_k + s_k - g) \\ f \end{bmatrix} \quad (2.2.33)$$

The second and the third step of indirect ADMM can be carried straight forward. In summary, for indirect ADMM, the iteration consists of following steps

$$1. \begin{bmatrix} H + \rho G^T G & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} z_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} -\rho G^T(\tau_k + s_k - g) \\ f \end{bmatrix} \quad (2.2.34)$$

$$2. s_{k+1} = \max\{0, -Gz_{k+1} + g - \tau_k\} \quad (2.2.35)$$

$$3. \tau_{k+1} = \tau_k + Gz_{k+1} + s_{k+1} - g \quad (2.2.36)$$

2.3 Saddle point type problem and the null space method

IPM-based MPC algorithms presented in [16] (primal-dual IPM) and [8] (primal IPM) require to solve a system of linear equation at the iteration j ($j = 1, 2, \dots$) with a general form as the following

$$\begin{bmatrix} E & F^T \\ F & O \end{bmatrix} \begin{bmatrix} * \\ * \end{bmatrix} = \begin{bmatrix} * \\ * \end{bmatrix} \quad (2.3.37)$$

where E is positive definite matrix and changes every iteration.

In [8, 16], different methods exploiting structure of sparse MPC formulation for solving this system of equations were proposed. These methods are based on the Schur reduction

and block-wise Cholesky factorisation with different implementation strategies.

As shown above, the first step of direct ADMM requires to solve a system of linear equations:

$$\begin{bmatrix} H + \rho I & F^T \\ F & O \end{bmatrix} \begin{bmatrix} * \\ * \end{bmatrix} = \begin{bmatrix} * \\ * \end{bmatrix} \quad (2.3.38)$$

where H is the Hessian of the sparse QP (see (2.1.11)) ρ is the step-size of ADMM.

Similarly, indirect ADMM requires a system of linear equations each iteration as

$$\begin{bmatrix} H + \rho G^T G & F^T \\ F & O \end{bmatrix} \begin{bmatrix} * \\ * \end{bmatrix} = \begin{bmatrix} * \\ * \end{bmatrix} \quad (2.3.39)$$

Unlike (2.3.37), the left-hand matrix in (2.3.38) is fixed at every iteration if the step-size ρ is fixed. Hence, one can pre-compute its inversion, and the solving of (2.3.38) reduces to a matrix-vector multiplication. In [44], this approach is adopted to solve (2.3.38). This approach is preferred for FPGA implementation as the matrix-vector multiplication is highly parallel-able. However, the bandedness of involved matrices are destroyed. Note that the pattern of (2.3.39) is similar to (2.3.38). [43] only discussed the theoretical aspect of indirect ADMM without implementation details.

Fast dual forward-backward splitting (FDFB) of [36] needs to solve

$$\begin{bmatrix} H & F^T \\ F & O \end{bmatrix} \begin{bmatrix} * \\ * \end{bmatrix} = \begin{bmatrix} * \\ * \end{bmatrix} \quad (2.3.40)$$

which is similar to (2.3.39) and (2.3.38). [36] also inverted the left-hand side matrix in (2.3.40) before entering the iterations.

Eq. (2.3.37) to (2.3.40) have a same pattern which is called as saddle point type problem, see [9]. There are various methods for solving this type of system of linear equations [9]. Nullspace method for solving this problem is described next.

Null space method for saddle point problem

Consider a system of linear equations in a saddle point type as follows:

$$\begin{bmatrix} E & F^T \\ F & O \end{bmatrix} \begin{bmatrix} z \\ \theta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.3.41)$$

where $E \in \mathbb{R}^{n \times n}$ is positive semidefinite, $F \in \mathbb{R}^{m \times n}$ with $m \leq n$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$.

Assumption 2.3.1.

- F is full row rank
- E is positive definite in the null space of F ($\forall w \in \text{null space of } F, w^T E w > 0$)

Given Assumption 2.3.1, the system of linear equations (2.3.41) has unique solution [9]. It can be solved via null space method (see [9]) as follows

- Compute a basis Y for the null space of F , i.e. $FY = O$
- Compute a particular solution $\hat{z}$ of $Fz = \beta$, i.e. $F\hat{z} = \beta$
- Solve

$$Y^T E Y v = Y^T (\alpha - E \hat{z}) \quad (2.3.42)$$

- v_* is obtained from (2.3.42) and then $z_* = Y v_* + \hat{z}$.
- θ is obtained by solving

$$F F^T \theta = F(f - E z_*) \quad (2.3.43)$$

Using null space method, the original system is solved via solving two smaller system of linear equation as (2.3.42)-(2.3.43). Null space method is especially attractive when $n - m$ (dimension of matrix E versus (row) dimension of matrix F) is small [9]. In sparse MPC formulation, dimension of E is $N(n_x + n_u)$ and of F is $N n_x$ (n_x , n_u , N : number of states, number of control inputs, and control horizon). Hence, for MPC problem, null space method may be attractive when $N n_u$ is small.

The null basis Y is not unique and there are several choices, such as an orthogonal basis or a triangular basis, see [9]. For a banded matrix, there may exist a banded null basis [45]. To compute a banded basis, one can employ ‘turnback’ algorithms [46], originated by the structural engineering community.

Turnback algorithm to find banded null basis

The turnback method was first developed by [46], and its interpretation as a matrix factorization was given in [47]. Later extensions include [45] and [48]. Practical experiments confirmed that turnback algorithms are the most effective algorithms for constructing a banded null basis [45]. Basically, turnback algorithms have two phases. In the first phase, they search for a set of active columns $\{c_k\}$, where an active column c_k is a column such that the set of columns starting from 1 to c_k , excluding previously found active columns,

Figure 2.4: Control system with the event-triggered sequence-based anytime algorithm (E-SAC)

are linearly dependent. In the second phase, the turnback phase, the search is conducted in reserved direction. Then, the coefficients for the linear dependent columns form a vector of the null basis. Detail of turnback algorithm is presented later in Section 3.4.

In fact, for sparse MPC formulation, the equality constraint matrix is banded. However, a banded null basis has not been exploited in previous MPC-related optimisation algorithms. Moreover, the existence of a banded null basis in MPC has not been confirmed theoretically in the existing literature.

2.4 Event-triggered sequence-based anytime control

Event-triggered sequence-based anytime control belongs to the family of anytime control. Anytime control algorithm was first proposed in [49] to deal with time-varying processor availability. It uses an idea from the artificial intelligent (AI) community called *anytime algorithm* [50], which is a computational procedure that could provide a valid answer even when it is terminated prematurely.

There are various forms of anytime control algorithms. In [49], different controllers with different floating point operations were designed. Notable later works include [10, 51–53]. In [51], a stochastic switching law within a set of pre-designed controllers is proposed. In [52], the main idea is to sequentially calculate the components of the plant input vector. In [10], named as sequence-based anytime control (SAC) as aforementioned, a buffer is used to store the tentative future inputs. In [53], a method for co-design of estimator

and controller is proposed where the controller requests a time-varying criterion for the estimator. When sensor measurements are transmitted through a communication network, the measurements may be unavailable due to packet dropouts, or network congestion. Among anytime control algorithms, SAC can handle this situation since it has a buffer which serves to provide a control input even when no measurement is received.

Motivated by the idea of using event-triggered control (see e.g. [54–58]) as a method to reduce demands on the network and computing processor while guaranteeing satisfactory levels of performance [59], SAC with an event-triggering mechanism (E-SAC) was proposed in [11].

Fig. 2.4 shows an E-SAC-based control system. The computational resources, e.g. processor and memory availability, are time-varying. When they are available, E-SAC not only computes the current input but also computes a sequence of tentative inputs and saves them into a buffer for a “rainy day”. To reduce the demand on communication channel, the sensor measurement is transmitted to the controller using an event-triggered mechanism, e.g. when the measurement value exceeds a pre-determined threshold. Compared with a no-buffering algorithm, it is demonstrated in [11] that E-SAC achieves a better performance. Stochastic stability of E-SAC has been derived in [11] using the State-dependent Random-time Drift condition approach of [18], assuming that the computational resources are modelled as identical and independent distributed (i.i.d) random processes ⁱⁱ.

Currently, existing E-SAC only features one control law in the buffer. However, in control system design, one may encounter situations where one would like to switch between different control laws in respond to changing operating conditions. For example, one may switch between a suboptimal or optimal solution in MPC [12], a fixed- or floating-point controller implementation [13]. However, E-SAC with two or more control laws with different computational cost has not been considered in the literature. This thesis extends the existing E-SAC to schemes that feature multiple control laws, and develops a new stability analysis method based on Markov jump system. The new approach systematically extends to a more sophisticated E-SAC schemes for which, until now, no analytical expression had been obtained.

In the next chapters, details of the proposed methods in this thesis are presented.

ⁱⁱThe i.i.d assumption is not a restriction.

Chapter 3

Banded null basis and ADMM for embedded MPC

Summary: This chapter describes the proposed banded null basis methods to accelerate the solution of MPC. The advantage of the proposed method is demonstrated in two situations (1) solving the KKT system arising from ADMM and (2) transforming the sparse MPC formulation to a reduced-Hessian QP while maintaining the banded structure of the Hessian. The turnback algorithm is employed to find a banded null basis. The existence of the banded null basis in MPC is proved for single-input systems, and an upper bound for its sparsity density is derived. In case an ill-conditioned basis arises, methods for improving the conditioning are developed. The improved ADMM algorithms are implemented in an industrial device with a set of 15 MPC problems, and in the ARM-Cortex A9 processor with a quadrotor problem. The last section of this chapter introduces existing automatic C-code generation tools for embedded implementation of MPC.

Notation: For a vector v , $\|v\|_2$ denotes the 2-norm. $\mathbb{R}^{m \times n}$ denotes the set of real matrices with m rows and n columns. $\mathbb{R}^m$ denotes a real vector of dimension m . $\otimes$ denotes the Kronecker product. $\mathbb{N}$ represents the set of natural numbers. For a matrix Y , Y_i denotes the i -th column, and Y_{ij} denotes the entry at row i and column j . The indicator function of a set $\mathcal{G}$ is denoted as $\mathcal{I}_{\mathcal{G}}(\cdot)$ meaning that $\mathcal{I}_{\mathcal{G}}(e) = 0$ if $e \in \mathcal{G}$ and $\mathcal{I}_{\mathcal{G}}(e) = +\infty$ if $e \notin \mathcal{G}$.

3.1 Banded null basis to solve KKT system

In this section, banded null basis method is applied to solve the KKT system (2.2.23) and (2.2.34) arising from ADMM algorithms. Due to the similarity of (2.2.23) and (2.2.34), only the solution of (2.2.34) is presented. For ease of reference, system of linear equations

(2.2.34) is repeated here as:

$$\begin{bmatrix} H + \rho G^T G & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} z_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} d_k \\ f \end{bmatrix} \quad (3.1.1)$$

where $d_k = -\rho G^T(\tau_k + s_k - g)$.

Let Y be a basis of the null space of F (F is defined in (2.1.10)), i.e. $FY = O$ where O is a zero matrix with appropriate dimensions. Following (2.3.42)-(2.3.43), KKT system (3.1.1) is solved as follows:

$$1. Y^T(H + \rho G^T G)Y v_{temp} = \mu_k \quad (3.1.2)$$

$$2. z_{k+1} = \bar{z} + Y v_{temp} \quad (3.1.3)$$

$$3. \text{ No need to calculate } \theta_{k+1} \text{ as it is not involved in} \quad (3.1.4)$$

later ADMM steps

where $\mu_k = Y^T(-d_k - (H + \rho G^T G)\bar{z})$, $\bar{z}$ is a particular solution of $Fz = f$, i.e. $F\bar{z} = f$; $d_k = \rho G^T(\tau_k + s_k - g)$, and v_{temp} is a new variable which is used to parametrise $Fz = f$, i.e. $z = \bar{z} + Y v_{temp}$.

To compute a particular solution $\bar{z}$, recall that for sparse MPC formulation, $Fz = f$ with $z = (u_0; x_1; u_1; x_2; \dots; u_{N-1}; x_N)$ (see (2.1.6)) and $f = [Ax_0; 0; \dots; 0]$ (see (2.1.8)), is equivalent to set of following equations:

$$\begin{aligned} x_1 &= Ax_0 + Bu_0, \quad x_2 = Ax_1 + Bu_1, \quad \dots, \\ x_N &= Ax_{N-1} + Bu_{N-1} \end{aligned}$$

Then a particular solution of $Fz = f$ is any $\bar{z} = (u_0; x_1; u_1; x_2; \dots; u_{N-1}; x_N)$ whose elements satisfy the above set of equations. One such particular solution is: $u_0 = u_1 = \dots = u_{N-1} = 0$ and $x_1 = Ax_0, x_2 = Ax_1, \dots, x_N = Ax_{N-1}$.

The null basis Y is not unique, and there are several choices such as an orthogonal null basis or a triangular null basis, see [9]. Since F is a banded matrix, there may exist a banded null basis for its null space [45]. To compute a banded null basis, the turnback algorithm of [45] can be employed. This thesis exploits the structure of the equality constraint matrix to propose a more efficient method for computing the banded basisⁱ. Once a banded null basis Y is obtained, the matrix $\bar{H} = Y^T(H + \rho G^T G)Y$ on the left side of (3.1.2) is a banded matrix (see e.g. in Fig. 3.1 the centre image). Equation (3.1.2) is then solved by offline computing the Cholesky factorization of $\bar{H}$ (as this matrix is fixed

ⁱThe turnback algorithm, the new proposed approach and the existence of a banded null basis in MPC are described in Section 3.4.

Figure 3.1: Pattern of $Y^T(H + \rho G^T G)Y$ in (3.1.2) (center sub-figure) and its Cholesky factorisation matrix (right figure, the "lower" form of Cholesky factorisation is used) (MPC problem with 6 states & 2 inputs & control horizon 10)

for LTI system), and then carrying out the backward and forward substitutions.

Compared with the original system (3.1.1), system of linear equations (3.1.2) has a smaller dimension while still maintaining the banded structure. This is the main reason that the solution of (3.1.1) can be accelerated. Implementation results showed in later section confirm the efficiency achieved by banded null basis method.

Remark 3.1.1. In this thesis, only linear time-invariant (LTI) systems are considered. The design parameters of the MPC problems are assumed to be fixed. Hence, all matrices are fixed. Then, the banded null basis, as well as the Cholesky factorisation matrix are computed offline (i.e. the pre-processing step). For time-varying systems, the banded null basis Y and the Cholesky factorisation of $Y(H + \rho G^T G)$ need to be recomputed each sampling instant.

3.2 Banded reduced-Hessian QP problem

In this section, banded null basis method is applied to obtain a banded reduced-Hessian QP.

Consider the sparse QP problem (2.1.7), by parametrising z as $z = \bar{z} + Yy$, where $\bar{z}$ is a particular solution, Y is a basis and y is a new decision variable, and eliminating the equality constraints $Fz = f$, the following equivalent reduced-Hessian QP with reduced

Figure 3.2: Matrices in the banded reduced-Hessian QP

decision variables is obtained:

$$\min_y \quad \frac{1}{2} y^T \tilde{H} y + (\bar{z}^T H_0) y \quad (3.2.5)$$

$$\text{s.t.} \quad \tilde{G} y \leq g - G \bar{z} \triangleq \tilde{g} \quad (3.2.6)$$

where $\tilde{H} = Y^T H Y$ (reduced Hessian), $\tilde{G} = G Y$, $H_0 = H Y$.

If Y is a banded null basis, the reduced Hessian matrix $\tilde{H}$, $\tilde{G}$ and H_0 are usually banded matrices. Fig. 3.2 shows the banded pattern of the reduced Hessian QP for the spring-mass system problem of [60].

The number of decision variables of QP problem (3.2.5)-(3.2.6) is $N n_u$, i.e., $y \in \mathbb{R}^{N n_u}$, whereas it is $N(n_x + n_u)$ in the original QP (2.1.7). The number of decision variables of the reduced-Hessian QP problem is the same as in the condensed QP showed in Section 2.1. If $n_u \ll n_x$, the reduction in the number of decision variables can be quite substantial. For ease of reference, the condensed QP is recalled:

$$\min_{\theta} \quad \frac{1}{2} \theta^T H_c \theta + (x_0^T \Gamma_1) \theta \quad (3.2.7)$$

$$\text{s.t.} \quad \Gamma_2 \theta \leq g_c + [\Gamma_3 x_0; 0_{N n_u}] \triangleq \tilde{g}_c \quad (3.2.8)$$

where H_c , $\Gamma_1, \Gamma_2, \Gamma_3$ are shown equation (2.1.14).

The QPs (3.2.5) and (3.2.7) have a typical structure including a quadratic objective function and linear inequality constraints. Slack variables ν are added to (3.2.6) and (3.2.8) yielding a standard ADMM setup for (3.2.5) and (3.2.7). The following is the ADMM

iteration for solving QP (3.2.5):

$$1. y_{k+1} = -(\tilde{H} + \rho\tilde{G}^T\tilde{G})^{-1}[H_0^T\bar{z} + \rho\tilde{G}^T(\nu_k + \gamma_k - \tilde{g})] \quad (3.2.9)$$

$$2. \nu_{k+1} = \max\{0, -\tilde{G}y_{k+1} + \tilde{g} - \gamma_k\} \quad (3.2.10)$$

$$3. \gamma_{k+1} = \gamma_k + \tilde{G}y_{k+1} + \nu_{k+1} - \tilde{g} \quad (3.2.11)$$

where γ_k is the Lagrangian multiplier.

Banded matrices are exploited in (3.2.9)-(3.2.11) as follows. In (3.2.9), instead of calculating the inversion $(\tilde{H} + \rho\tilde{G}^T\tilde{G})^{-1}$ and doing the matrix-vector multiplication, y_{k+1} in (3.2.9) is calculated by solving

$$(\tilde{H} + \rho\tilde{G}^T\tilde{G})y_{k+1} = [H_0^T\bar{z} + \rho\tilde{G}^T(\nu_k + \gamma_k - \tilde{g})] \quad (3.2.12)$$

In this system of linear equation, $\tilde{H} + \rho\tilde{G}^T\tilde{G} \triangleq \tilde{H}_1$ is a banded matrix (same pattern as $\tilde{H}$ showed in Fig. 3.2). Its Cholesky factorization is computed offline, and (3.2.12) is solved via forward and backward substitutions. The right-hand side of (3.2.12) and the iterations (3.2.10)-(3.2.11) contain only matrix-vector multiplications in which exploiting banded matrices is straightforward.

The ADMM iteration for the condensed QP (3.2.7) can be adapted directly from (3.2.9)-(3.2.11) by replacing $\tilde{H}$ by H_c , etc. However, since matrices in the condensed formulation are dense except for Γ_2 (in some cases), the calculation of (3.2.9)-(3.2.11) with the set of matrices from condensed QP (3.2.7) will be more costly than the reduced Hessian (3.2.5). It can be said that, the banded null basis method gives ADMM “good ingredients” so that ADMM can run more efficiently.

Remark 3.2.1. By simple algebraic manipulations, one can show that ADMM for the reduced Hessian QP and indirect ADMM will give the same solution, given the same initial condition, the same number of iterations and the same step-size. The differences in two methods are in the set of matrices, and the dimension of the primal decision variables, i.e. z_k versus y_k .

Remark 3.2.2. Note that after transforming the sparse QP to a reduced-Hessian QP, the box constraints of the original QP (if exist) will become polytopes in the new QP.

3.3 Schur complement reduction for KKT system

Schur complement reduction has been know as an effective method for solving the KKT system in interior-point-based MPC optimisation algorithms, see e.g. [8] [16]. The KKT system (3.1.1) has the same structure as the structure of KKT matrix in the barrier

Figure 3.3: Pattern of $F\tilde{H}^{-1}F^T$ in (3.3.13) (left figure) and its Cholesky factorisation matrix (right figure, the "lower" form of Cholesky factorisation is used) (MPC problem with 6 states & 2 inputs & control horizon 10)

interior-point method of [8]. Then, the method in [8] is adopted to solve (3.1.1). Implementation results in later section show that banded null basis outperforms, up to 7.8 times faster, Schur complement reduction technique.

In the following, the steps required to solve (3.1.1) using Schur complement reduction are presented.

Using Schur complement reduction [9], (3.1.1) is reduced to

$$1. F\tilde{H}^{-1}F^T\theta_{k+1} = -f + F\tilde{H}^{-1}d_k \quad (3.3.13)$$

$$2. z_{k+1} = -\tilde{H}^{-1}(F^T\theta_{k+1} - d_k) \quad (3.3.14)$$

where $d_k = \rho G^T(\tau_k + s_k - g)$ and

$$\tilde{H} = (H + \rho G^T G)$$

Since matrix $\tilde{H}$ is block diagonal, then $\tilde{H}^{-1}$ is also block diagonal; and $F\tilde{H}^{-1}F^T$ in (3.3.13) has a "step" pattern (see Fig. 3.3). Given an LTI system and a fixed step-size ρ , $F\tilde{H}F^T$ is a fixed matrix. Then, its Cholesky factorisation can be computed offline. Note that only the non-zero elements of the Cholesky factorisation matrix need to be stored. Then (3.3.13) is solved by backward and forward substitutions.

Exploiting structure to carry out (3.3.14) is quite straightforward, since the computation is mainly matrix-vector multiplication.

Remark 3.3.1. Both Schur reduction method and banded null basis method result in a system of linear equations with banded matrices, see (3.1.2) and (3.3.13). These systems of linear equations are solved by the same method, i.e. Cholesky factorization. It can be seen that the Cholesky factorisation matrix of the banded null basis method has a smaller

dimension and fewer non-zero elements than that of the Schur complement reduction method (dimension 20 vs 60, non-zero elements 119 vs 534) in a particular MPC problem showed in Fig. 3.3 and 3.1. As a result, the banded null basis method requires fewer multiplications when carrying out the backward and forward substitutions than that of the Schur complement reduction method. This is the main reason that the banded null basis method outperforms Schur reduction method in all benchmark problems in this thesis.

3.4 On the computation and the existence of a banded null basis in MPC

This section introduces methods for computing a banded null basis in MPC, including the turnback algorithm of [45] and a new proposed algorithm. There is no theoretical result to confirm whether turnback always returns a banded null basis, as well as no theoretical result on the existence of banded null basis in MPC. This thesis proves this.

After introducing the turnback algorithm as below, a new algorithm for computing a banded null basis for a single-input system is presented. In addition, the banded null basis exists (with certain condition) as a results of the termination in finite steps of the new proposed algorithm.

3.4.1 Turnback algorithm

The turnback method was first developed by [46], and its interpretation as a matrix factorization was given in [47]. Later extensions include [45] and [48]. Practical experiments confirmed that turnback algorithms are the most effective algorithms for constructing a banded null basis [45].

Algorithm 3.4.1 below describes turnback algorithm of [45]. Basically, it has two phases. In the first phase, the algorithm searches for a set of active columns $\{c_k\}$, where an active column c_k is a column such that columns starting from 1 to c_k , excluding previously found active columns, are linearly dependent. In the second phase, the turnback phase, starting from an active column c_k , the search is conducted in the reserved direction to find a turnback column t_k . Then, step 2.3 in the Algorithm 3.4.1 below forms the null basis.

Algorithm 3.4.1: Turnback algorithm.

Given $F \in \mathbb{R}^{m \times n}$ ($m < n$), we want to construct a banded null basis $Y \in \mathbb{R}^{n \times r}$ ($r = n - m$), i.e $FY = 0_{m \times r}$. Denote F_i , $i = 1, 2, \dots, n$, the i -th column. Given S a subset of

$\{1, 2, \dots, n\}$, F_S is a set of columns of F whose indices are in S , $Y_i(S)$ are elements in i -th column of Y whose indices are in S . The main steps are as follows:

1. Find active indices $c_1, c_2, \dots, c_r$ so that the set of columns $\{F_1, F_2, \dots, F_{c_k}\} \setminus \{F_{c_1}, \dots, F_{c_{k-1}}\}$ are linearly dependent ($k = 1, 2, \dots, r$ is iteratively increased to find a corresponding c_k , $\{F_{c_1}, \dots, F_{c_{k-1}}\} = \emptyset$ if $k = 1$)
 2. Turn-back step:
 - 2.1. $k = 1$, $\mathcal{T} = \emptyset$
 - 2.2. Set column F_{c_k} as active column
 - 2.3. Find $t_k = \max\{j | j < c_k \& F_{S_k} \triangleq \{F_{c_k}, F_{c_k-1}, \dots, F_j\} \setminus \{F_{\mathcal{T}}\} \text{ are linearly dependent}\}$ ($S_k = \{c_k, c_k - 1, \dots, j\} \setminus \mathcal{T}$ is a set of indices in descending order). Find linear dependent coefficients vector b_k for F_{S_k} : $F_{S_k} b_k = 0$ & the last element of b_k is set to 1. Assign $Y_k(S_k) = b_k$
 - 2.4. $k = k + 1$, $\mathcal{T} = \mathcal{T} \cup \{t_k\}$. If $k \leq r$, return to 2.2, else quit and return Y .
-

This turnback algorithm has a mechanism to ensure that the computed basis is full rank.

3.4.2 New algorithm for efficiently computing the banded null basis, and the existence of banded null basis in MPC

It has been claimed, but without proof, that the turnback algorithm will always return a banded null basis in [45]. However, for the MPC problem, the equality constraint matrix F has a pattern as (2.1.10)). This thesis considers a *single-input* system as a starting point to present a new method of computing a banded null basis for MPC problems. This thesis also proves that the proposed algorithm always returns a banded null basis. More than that, an upper bound for the number of non-zero entries on each column and an upper bound for the sparsity density are derived. Details are presented as follows.

Consider system (2.1.5) with single control input, i.e. $n_u = 1$ and $B \in \mathbb{R}^{n_x \times 1}$. Recall that $A \in \mathbb{R}^{n_x \times n_x}$ (n_x : number of states)

For ease of reference, equality constraints matrix F of the sparse MPC formulation is re-called:

$$F = \begin{bmatrix} -B & I & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -A & -B & I & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & -A & -B & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -A & -B & I \end{bmatrix} \quad (3.4.15)$$

Algorithm 3.4.2 (named as Fast Computation of Banded Null Basis (FCBNB)):

To compute a banded basis Y for the equality constraint matrix F in MPC problem (single-input system), i.e. solve $FY = O = F[Y_1 \ Y_2 \ \dots \ Y_{Nn_u}]$

Input: A, B, N (control horizon)

1. Find the smallest number Ω such that $B, AB, \dots, A^\Omega B$ are linearly dependent columns (i.e. $\Omega = \text{rank of controllability matrix}, \Omega \leq n_x$). Then, there exist the set of control $\{u_k\}_{k=0,1,\dots,\Omega}$ such that $Bu_\Omega + ABu_{\Omega-1} + \dots + A^\Omega Bu_0 = 0$ and $u_0 = 1$.
2. Assume that $\Omega < N$, and select:

$$Y_1 = [u_0; x_1; u_1; x_2; \dots; u_\Omega; x_{\Omega+1}; 0_u; x_{\Omega+1}; 0_x; \dots; 0_u; x_N] \quad (3.4.16)$$

where

$$x_0 = 0$$

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, \dots, \Omega$$

$$x_{k+1} = Ax_k, \quad k = \Omega + 1, \dots, N$$

$$0_u = \text{zero vector with dimension } n_u$$

$$0_x = \text{zero vector with dimension } n_x$$

Due to step (1), and since $x_k = A^k x_0 + \sum_{i=0}^k A^{k-1-i} Bu_i$, it follows that $x_{\Omega+1} = x_{\Omega+2} = \dots = 0$ and hence

$$Y_1 = [u_0; x_1; u_1; x_2; \dots; u_\Omega; 0_x; 0_u; \dots; 0_u; 0_x]$$

3. Select

$$Y_2 = [0_u; 0_x; u_0; x_1; \dots; u_\Omega; 0_x; \dots; 0_u; 0_x]$$

$$Y_3 = [0_u; 0_x; 0_u; 0_x; u_0; x_1; \dots; u_\Omega; 0_x; \dots; 0_u; 0_x]$$

....

$$Y_{N-\Omega} = [0_u; 0_x; \dots; u_0; x_1; \dots; x_{\Omega-1}; u_\Omega; 0_x]$$

4. To find remaining columns of the basis Y :

Denote $E_k, k = 1, 2, \dots, n_x$, as the k -th column of the identity matrix whose dimension is $n_x \times n_x$, i.e.

$$E_k \triangleq \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix}^T \in \mathbb{R}^{n_x}, \quad (3.4.17)$$

$k \text{ elements}$

- 4.1. Form the matrix

$$C_\Omega = [A^{\Omega-1}B \dots AB \ B \ E_1 \dots E_{n_x}]$$

Perform the QR factorization of C_Ω to obtain a set of columns $\{E_k\}$ that have a “zero pivot”. These are the so-called “active column”, i.e. they are linearly dependent with the columns on their left.

Set $\mu = 0$ (counting variable)

- 4.2. For each active column E_k : find the smallest number $0 \leq \Psi_k < \Omega$ such that E_k can be represented as a linear combination of $E_{k-1}, \dots, E_1, B, AB, \dots, A^{\Psi_k-1}B$ (by sequentially forming $C_i = [A^iB, \dots, AB, B, E_1, \dots, E_k]$ and carrying the Gauss-Jordan elimination to obtain the reduced row echelon form; then checking if the last column contains a pivot (meaning independence) or not (meaning $\Psi_k = i$). Then:
 - 4.2.1. Solve for a set of scalar coefficients $\{\epsilon_k, \epsilon_{k-1}, \dots, \epsilon_1, u_{\Psi_k}, u_{\Psi_k-1}, \dots, u_0\}$ such that $\epsilon_k E_k + \dots + \epsilon_1 E_1 + Bu_{\Psi_k} + ABu_{\Psi_k-1} + \dots + A^{\Psi_k}Bu_0 = 0$ and $u_0 = 1$.
 - 4.2.2. $\mu := \mu + 1$ (counting variable)
 - 4.2.3. Choose

$$Y_{N-\Omega+\mu} = [\underbrace{0_u; 0_x; \dots; 0_x}_{((N-\Psi_k-1)(n_x+n_u))} \ ; \ u_0; x_1; u_1; \dots; x_{\Psi_k-1}; u_{\Psi_k}; x_{\Psi_k+1}]$$

where $x_{i+1} = Ax_i + Bu_i$, $i = 0, 1, \dots, \Psi_k$; $x_0 = 0$. One can also show that $x_{\Psi_k+1} = [-\epsilon_1 \dots -\epsilon_k \ 0 \dots 0]^T$ due to Step 4.2.1

Assumption 3.4.1. $\Omega < N$ (controllability index is smaller than the control horizon).

This assumption implicitly implies that the control horizon should be large enough. A large control horizon allows the sparsity of sparse MPC formulation to be showed off. Moreover, choosing a large control horizon N is a sufficient condition to guarantee the stability of MPC in some cases, e.g., see [61].

Theorem 3.4.1 (Single-input case ($n_u = 1$)). Assume that Assumption 3.4.1 holds, Algorithm 3.4.2 returns a full column-rank banded basis Y . The number of non-zero elements on each column does not exceed $\Omega(n_x + n_u) + n_u$, and the total of the Euclidean distances, with respect to the diagonal line (see illustration in Fig. 3.4), of the first and last non-zero elements of each column does not exceed

$$\chi = \frac{\Omega + \frac{n_u}{n_x+n_u}}{\sqrt{1 + \left(\frac{n_u}{n_x+n_u}\right)^2}} \quad (3.4.18)$$

Figure 3.4: The diagonal line of the banded basis passing through the left-top and right-bottom corners; point P : last non-zero element of column Y_2 ; red arrow: the Euclidean distance from P to the diagonal line.

where Ω is the controllability index of (A, B) .

Proof:

It is obvious that Step 1 to 3 of Algorithm 3.4.2 always returns $N - \Omega$ columns, column Y_1 to column $Y_{N-\Omega}$. One can show that column Y_1 to column $Y_{N-\Omega}$ are linearly independent, since the positions of their first non-zero element are different.

To proof that Step 4 returns the remaining Ω columns of the basis, following lemmas will be proved.

Lemma 3.4.1. There exist exactly Ω columns within the set of columns $\{E_k\}_{k=1,2,\dots,n_x}$ that satisfy the “active column” condition in step 4.1 of Algorithm 3.4.2.

Proof: See Appendix 3.10.1.

Lemma 3.4.2. $\Psi_k < \Omega$ as stated in Step 4.2 of Algorithm 3.4.2 exists $\forall k$ when E_k is an active column. In addition, $\Psi_k \neq \Psi_j \forall k \neq j$.

Proof: See Appendix 3.10.2.

Lemma 3.4.1 and Lemma 3.4.2 imply that Step 4 returns Ω columns of basis Y . Lemma 3.4.2 implies that the position of the first non-zero element of columns $Y_{N-\Omega+\mu}$ ($\mu = 1, 2, \dots, \bar{\Omega}$) are $(\Psi_k + 1)(n_x + n_u)$ counted from the last-row element. One can show that these positions are different (due to $\Psi_i \neq \Psi_j$). In addition, by the fact that $\Psi_k < \Omega$, the positions of first non-zero elements of last Ω columns differ from these of the first $N - \Omega$ columns (whose the largest position index of the first non-zero element is $(\Omega+1)(n_x+n_u)+1$ counted from the last-row element, and $(\Omega+1)(n_x+n_u)+1 > (\Psi_k+1)(n_x+n_u) \forall k$).

Therefore, since the positions of first non-zero element on each column are different, Y is full column-rank.

The Euclidean distances from the first and the last non-zero elements to the diagonal line of the basis can be calculated using geometry laws (cosine law and Pythagorean law, see Fig. 3.4 for illustration of the Euclidean), and their summation is bounded by χ (eq. (3.4.18)). The proof is then finished. ■

Remark 3.4.1. From Theorem 3.4.1 and its proof, it follows that:

- The bandwidth of the obtained banded null basis depends on the controllability index of the system and is independent of the prediction horizon.
- For a general matrix, turnback algorithm is computational demanding as it requires repeated LU (or QR) decompositions to find active and turnback columns. However, for matrix F arising from MPC, the computation of the banded null basis is significantly less computational demanding by using the new proposed algorithm.
- Step 4 of Algorithm 3.4.2 follows the principle of turnback algorithm which carrying two phases. The difference is in the way the linear coefficients are computed in Step 4.2.1. In turnback algorithm, it ignores columns $\{A^{\Psi_i}B\}$ that have been found, whereas Algorithm 3.4.2 does not do so. However, the coefficients of these columns can be 0 in Step 4.2.1, and then Step 4.2.1 gives the same result as turnback algorithm does.
- Algorithm 3.4.2 gives the same result as the turnback algorithm does, except for the columns $Y_{N-\Omega+1}$ to Y_N (computed by Step 4 of Algorithm 3.4.2). However, the sparsity difference is negligible.

When extending Algorithm 3.4.2 to multi-input system, the computation cannot be extended straight-forward due the differences in the controllability indices. However, numerical examples have been carried out and no countered result occurred. Hence, the following conjecture is made.

Conjecture 3.4.1. Assume that Assumption 3.4.1 holds, then there exist a banded null basis Y of MPC problem. The number of non-zero elements on each column does not exceed $\Omega(n_x + n_u) + n_u$, and the total of the Euclidean distances, with respect to the diagonal line (see illustration in Fig. 3.4), of the first and last non-zero elements of each column does not exceed

$$\chi = \frac{\Omega + \frac{n_u}{n_x + n_u}}{\sqrt{1 + \left(\frac{n_u}{n_x + n_u}\right)^2}} \quad (3.4.19)$$

where Ω is the controllability index of (A, B) .

3.5 Improving the condition number of the banded null basis

A challenge arising is that the banded null basis may be ill-conditioned. The turnback algorithm is tested in 18 MPC examples available in [62]ⁱⁱ, as well as a large number of random problems. Results showed that in some of these MPC problems, the banded null bases are very ill-conditioned which may cause numerical issues such as serious computation error in low precision arithmetic.

To avoid ill-conditioned banded bases, methods for improving the conditioning of the banded null basis are proposed in this section.

Method 1 - Column equilibrating

A simple technique for tuning the conditioning of an arbitrary null basis is first proposed. For $Y \in \mathbb{R}^{n \times r}$ a null basis of $F \in \mathbb{R}^{m \times n}$, $r = n - m$, $\tilde{Y} = YD$ is also a null basis for any arbitrary full-rank diagonal matrix $D \in \mathbb{R}^{r \times r}$. Denote the diagonal entries of D as $d_{ii}, i = 1, 2, \dots, r$. The columns of $\tilde{Y}$ are actually the scaled columns of Y , i.e., $\tilde{Y}_i = d_{ii}Y_i$. D is chosen to equilibrate the columns of Y , i.e., to make the norm of all columns of $\tilde{Y}$ the same. The equilibration in 2-norm is selected, and D is chosen as:

$$d_{ii} = \frac{1}{\sqrt{\|Y_i\|_2}}$$

which makes the 2-norm of all columns be equal to unity. Using Method 1, the sparsity of the basis remains the same as that of the original.

Method 2: Column combination

The idea is to replace each column of the banded null basis by a linear combination of itself and its immediate neighbour on the right. This is equivalent to replacing the basis Y by a new basis $\bar{Y} = YW$ where

$$W = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{21} & 1 & 0 & \cdots & 0 \\ 0 & w_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & w_{r(r-1)} & 1 \end{bmatrix} \quad (3.5.20)$$

ⁱⁱAlgorithm 3.4.2 and turnback algorithm return the same results in single-input systems among 18 considered MPC problems.

Thus, improving the condition number of a banded null basis can be posed as an optimisation problem as

$$\min_{w_{21}, w_{32}, \dots, w_{r(r-1)}} \text{cond}(YW) \quad (3.5.21)$$

where $\text{cond}(M)$ denotes the condition number of a matrix M .

Remark 3.5.1. One can show that $F\bar{Y} = F(YW) = (FY)W = OW = O$. This implies that $\bar{Y}$ is still a basis of the nullspace, since W is non-singular. Note that if Y is pre-multiplied by a matrix, e.g. $Y^* = WY$, the obtained matrix is no longer a basis of the null space of F .

Matrix equilibration technique is a heuristic for reducing conditioning of a matrix [63]. Basically, this approach tries to make the norm of all rows and all columns equal. As an example, one can choose appropriate matrices D, E and multiply to the left and the right of a matrix $\Psi \leftarrow D\Psi E$, see [64].

However, since not all matrices can be equilibrated [64], there may not exist $\{w_{i(i-1)}\}$ such that all the rows of Y have the same 2-norm. Instead, the following problem is solved

$$\min_{w_{i(i-1)}} J(\underline{w}) \triangleq \sum_{k=1}^n |r_k r'_k - \alpha| \quad (3.5.22)$$

where $r_k \in \mathbb{R}^{1 \times r}$ is the k -th row of $\bar{Y}$, ($k = 1, 2, \dots, n$), parametrized by set of parameters $\{w_{i(i-1)}\}_{i=2, \dots, r; j=1, 2, \dots, \eta} \triangleq \underline{w}$.

Problem (3.5.22) tries to make the 2-norm of each row of $\bar{Y}$ be as close to α as possible. The reason for choosing (3.5.22) is that it contains the quadratic function of optimisation variables $\{w_{i(i-1)}\}$. When $\alpha = 0$, $J(\underline{w})$ becomes convex and hence is favourable when online tuning of banded null basis is needed, especially for time-varying system.

The solution of (3.5.22) with $\alpha = 0$ is

$$w_{i(i-1)} = -\frac{\sum_{k=1}^n Y_{k(i-1)} Y_{ki}}{\sum_{k=1}^n Y_{ki}^2}, \quad i = 1, 2, \dots, r-1 \quad (3.5.23)$$

Remark 3.5.2. When $\alpha \neq 0$, problem (3.5.22) is non-convex. The effectiveness of choosing $\alpha \neq 0$ is a subject of current investigation.

Implementation results showed in Section 3.6 confirm that the proposed heuristic gives satisfactory results for the set of MPC benchmark problems. The proposed approach is

simple and can be used when online tuning the banded null basis is needed, since the computation of (3.5.23) is simple.

Example 3.5.1. Figure 3.5(a) shows the banded null basis Y of the equality constraint matrix F which arises from ATFI-16 aircraft MPC problem [1] [2] with a control horizon $N = 4$. Figure 3.5.b shows the new basis $\bar{Y}$ obtained by applying Method 2. As expected, the new basis $\bar{Y}$ has more number of non-zero entries on each column, and has more total number of non-zero elements than the original one (91 vs 80). Fig. 3.5.c shows the second column Y_2 and its right neighbour column Y_3 . Fig.3.5.d shows how Y_2 changes when it is combined with Y_3 .

Figure 3.5: Banded null basis Y for the null space of matrix F arising in MPC problem with plant model of [1] [2].

Method 2.A: Linear combination with restrictive criteria

Figure 3.5(c) shows that when Y_2 is replaced by a combination of Y_2 and Y_3 , the number of non-zero entries in Y_2 increase significantly from 13 to 18 (equivalent to 38 %), see Fig. 3.5(d). Whereas when replacing Y_1 by $Y_1 + w_{21}Y_2$, the number of non-zero entries in Y_1 only increase 1 from 13 to 14 (about 8%).

This observation suggests that if the pattern of two consecutive columns is quite different, there will be a significant increase in sparsity of the column in the left of these two columns. To avoid this, another variant of Method 2, referred to as Method 2A, is proposed and described next.

Each column Y_i ($i = 1, 2, \dots, r$) of the banded null basis will have non-zero entries starting from position t_i to c_i , where t_i and c_i are known when the banded basis is computed. Referring to example (3.5.1), for the first column Y_1 : $t_1 = 1$ and $c_1 = 13$, for the second

Figure 3.6: Hardware used for the implementation: EOR-3D (ARM-9, CPU 400MHz, RAM 64MB, embedded Linux)

column: $t_2 = 2$ and $c_2 = 14$. By comparing the lowest and highest position indexes of non-zero entries, i.e. (t_i, c_i) , of two consecutive columns, one can know how the pattern will change after carrying out the linear combination.

Thus, by examining the position of the first and last non-zero elements, denoted as $\{c_k, t_k\}_{k=1,2,\dots,r-1}$, column Y_i will be replaced, i.e. $Y_i \leftarrow Y_i + w_{(i+1)i}Y_{i+1}$, if the following conditions are satisfied.

$$\begin{cases} c_{i+1} - c_i = 1 \\ t_{i+1} > t_i, \end{cases} \quad (i = 1, 2, \dots, r-1) \quad (3.5.24)$$

Criterion (3.5.24) ensures that the number of non-zero entries of each column of Y increases by at most one, i.e., it can say that the bandwidth of the new Y will increase at most one compared with the original. However, this method will result in fewer tuning parameters w_{ij} than that of Method 2.

Implementation results showed in later section confirm that Method 2A achieves its objective.

Combination of Method 2 and 2A with Method 1

Once the new basis by using Method 2 or 2A is obtained, one can apply Method 1 to the new basis, since Method 1 does not change the sparsity of its input matrix.

3.6 Case study 1: a set of MPC problems tested in an industrial device

The turnback algorithm implemented in MATLAB returned a banded null basis Y in all 18 MPC problems publicly available in [62] (See Fig. 3.9). These banded bases are used

to solve the KKT system and to obtain a banded-reduced Hessian.

Among these 18 MPC problems available in [62], 13 problems are taken for embedded implementation (problem 6,7,10,11,12 are taken out, since problems 6,7 and 12 have the same dimension as problem 2 (2×1), problem 10 has the same dimension as problem 1 (4×2), problem 11 has dimension 3×1 which is small and ignored). In addition, for more diversity in the problem dimensions, two additional MPC problems arising from using the Multiplexed MPC strategy [65] are taken. Totally, there are 15 MPC problems tested.

For each of the 15 MPC problems, a sparse QP formulation is formed, and then is solved using these methods:

- Direct ADMM (consisting of three steps as equations (2.2.23)-(2.2.25)). Three methods of solving the KKT system (2.2.23) are compared resulting in three versions of direct ADMM:
 - (1) Banded null basis method (see Section 3.1, the proposed technique in this thesis): this implementation is denoted as “B.b”
 - (2) Schur reduction method (see Section 3.3): this implementation is denoted as “Schur.”
 - (3) Standard implementation: by inverting the KKT matrix offline, online computation becomes a matrix-vector multiplication for the first step. This implementation is denoted as “Std.”
- Indirect ADMM (2.2.34)-(2.2.36), with three versions similar to Direct ADMM as above.
- Reduced variables techniques, with two versions:
 - Banded reduced-Hessian QP (proposed in this thesis, Section 3.2): the banded reduced-Hessian QP is solved by an ADMM algorithm
 - Condensed QP: The condensed QP is solved by an ADMM algorithm that is similar to the ADMM algorithm used in the banded reduced-Hessian QP above, but the set of matrices is different (see Section 3.2 for details)

Thus, there are 8 algorithms in total. These algorithms are written in MATLAB m-code, and are then converted to C-code by MATLAB Coder R2016a. The C-code of each algorithm is compiled (IDE: Eclipse Mars 2.0 in Debian 8, cross-compiler: arm-linux-gnueabi-gcc), and downloaded to an industrial device (ARM9 processor, 400Mhz, 64MB RAM, single core, and embedded Linux OS) named as EOR-3D shown in Fig. 3.6 (from A. Eberle GmbH).

Figure 3.7: Sub-plot 1 and 2: ratio of time-per-iteration of banded null basis method (B.b) versus two other methods (standard and Schur reduction). Sub-plot 3: Ratio of time-per-iteration of reduced-Hessian QP using banded null basis versus using condensed formulation. Vertical axis: ratio $\frac{n_x}{n_u}$.

Table 3.1 shows the runtime of 8 algorithms for the 15 considered MPC benchmark problems.

For different problem dimensions, the speedup obtained are different. From Table 3.1, it seems that the speedup depends on $\frac{n_x}{n_u}$, the ratio of state dimension to number of control inputs. Fig. 3.7 shows the speedup with respect to $\frac{n_x}{n_u}$.

The left-most sub-plot in Figure 3.7 shows that, for direct ADMM, the banded null basis method is at least 1.6 times and up to 26 times faster than the standard implementation (triangle point); and it is at least 1.5 and up to 7.8 times faster than the Schur reduction approach (circle points). In indirect ADMM, see the middle sub-plot in Figure 3.7, speedup with respect to the standard method and Schur reduction method are in range 2 to 12.9 and 1.29 to 2.08, respectively.

The right-most subplot of Fig. 3.7 shows that when applying ADMM to the banded reduced Hessian, runtime is up to 18 times faster than applying ADMM to the condensed QP formulation.

Fig. 3.8 shows the convergence property of direct ADMM for MPC problem 5, namely the double inverted pendulum. It shows that, when using single precision arithmetic, an ill-conditioned banded null basis (with condition number equal to 10^{12} in this MPC problem) causes significant numerical errors (yellow square points on the top of the figure). In this case, the solution is far from the correct answer even when sufficient amount of time is spent. Even when using double precision, the null basis was ill-conditioned which resulted

Table 3.1: Time (ms) per iteration of ADMM with different implementation methods

Sys. des.			Direct ADMM			Indirect ADMM			Reduced variable	
Sys.	$n_x \times n_u$	N	Std.	Schur.	b.b	Std.	Schur.	b.b	condense	b.b
1	4x2	10	2.37	1.13	0.67	3.11	1.47	1.03	5.00	0.74
		20	9.36	2.35	1.29	11.54	3.04	2.11	17.78	1.54
2	2x1	10	0.65	0.38	0.25	1.06	0.62	0.48	2.23	0.35
		20	2.27	0.76	0.49	3.62	1.24	0.96	7.79	0.71
3	11x3	10	12.93	6.06	2.61	11.80	5.75	2.66	1.66	1.32
		20	62.99	12.36	5.80	57.52	11.88	5.94	4.49	2.97
4	4x1	10	1.70	1.01	0.50	1.52	0.94	0.46	0.37	0.21
		20	6.58	2.07	1.06	6.03	1.97	0.99	0.86	0.46
5	4x2	10	2.38	1.16	0.67	2.42	1.17	0.73	0.93	0.45
		20	9.34	2.39	1.42	9.35	2.40	1.52	2.35	0.95
8	6x2	10	4.13	1.87	0.72	6.58	2.99	1.82	10.57	1.01
		20	17.73	3.88	1.50	25.63	6.19	3.78	38.92	2.31
9	56x3	10	328.00	109.28	13.93	455.93	169.55	73.79	116.30	23.73
		20	1230	223.76	46.6	1744	342	164.50	446.37	87.02
13	12x4	10	16.52	5.87	3.21	25.91	9.44	6.73	40.55	5.42
		20	87.35	12.02	7.07	116.72	19.05	14.18	150.59	11.87
14	4x2	10	2.37	1.06	0.67	3.80	1.76	1.34	7.71	1.00
		20	9.31	2.18	1.31	13.57	3.55	2.68	28.29	2.01
15	9x3	10	#	#	#	13.15	4.71	2.27	10.74	1.58
		20	#	#	#	59.60	9.57	4.77	38.85	3.36
16	7x4	10	7.19	2.66	1.49	11.33	4.33	3.17	25.60	2.44
		20	33.69	5.53	3.16	47.31	8.85	6.54	97.08	5.23
17	6x2	10	4.16	2.06	1.04	6.54	3.15	2.11	24.69	1.70
		20	17.92	4.29	2.17	25.48	6.48	4.37	39.65	3.63
18	6x3	10	5.38	2.28	1.29	5.39	2.32	1.42	1.66	0.91
		20	21.54	4.70	2.71	21.60	4.80	3.01	4.53	1.91
19	12x1	10	10.32	5.29	1.01	16.92	8.54	4.22	9.21	1.64
		20	50.2	10.89	2.36	70.62	17.25	8.69	4.53	1.91
20	6x1	10	3.19	1.7	0.57	5.28	2.73	1.57	5.29	0.79
		20	12.78	3.50	1.16	18.81	5.60	3.20	18.76	1.71

1. F16 aircraft, 2. Ball balancing, 3. Binary distillation column, 4. DC motor, 5. Double inv. pendulum, 8. Quansar helicopter, 9.10-mass chain, 13. Quadcopter, 14. Two-link robot arm, 15. Shell oil fractionator, 16. Spacecraft, 17. Spring-mass, 18. Triple inv. pendulum, 19. MMPC of 13, 20. MMPC of 8

in significant computation errors (red-triangle points).

Schur reduction also results in big computation errors (star points) compared with the standard implementation (dotted line), in single precision. The reason is that the matrices in the system of linear equations arising from the Schur reduction method has a big condition number (about 10^4 for this particular example).

Using the proposed heuristics to improve the condition number of the banded null basis, the condition number of the basis is reduced significantly to 113. As a result, the computation error of the banded null basis approach in single precision is reduced substantially (yellow-square points reduces to purple-cross points). In double precision, the

Figure 3.8: Convergence in terms of runtime (of direct ADMM) for MPC problem 5 (Double inverted pendulum). Banded null basis method (with condition tuning) in double precision approaches the “fine” sub-optimal solution fastest.

improved banded null basis (blue-circle points) gives comparable accuracy compared with the standard implementation (blue line) and double precision Schur reduction method (red-diamond points) but converges in shorter time to the desired solution.

In terms of time to obtain a “raw” sub-optimal solution whose Euclid distance to the solution is more than 10^{-3} , banded null basis (with condition number tuning) in single precision arithmetic requires the least time (purple cross points). In order to obtain a “fine” suboptimal solution whose Euclid distance to the optimal solution is less than 10^{-6} , the banded null basis (with condition number tuning) in double precision performs best (blue circle points). Moreover, the runtime of the double precision implementation of banded null basis is not much worst than a single precision implementation.

Reduction of conditioning using Method 2 and 2A

Among the 18 MPC problems in [62], there are 7 problems that have poorly conditioned banded bases. Only the tuning results of these 7 problems are reported in this section.

From Table 3.2, it is interesting that Method 1, the simplest method, performs quite well. However, condition numbers are still quite big.

Method 2 and 2A reduce the conditioning significantly (except for the Quansar helicopter problem). Method 2 outperforms Method 2A but the performance gap is not much. The combination of methods: M2+M1 and M2A+M1 gives further reduction especially for the double inverted pendulum problem (problem 5).

Table 3.3 shows the reduction of sparsity when using the tuning methods. Method 2

Figure 3.9: Banded null basis in 18 MPC problems ($N = 10$).

Table 3.2: Conditioning of banded null basis obtained using different methods (M2+M1: Appy Method 2 then apply Method 1. Similarly for M2A+M1.)

Name	Original cond.	Meth. 1	Meth. 2	Meth. 2A	M2+M1	M2A+M1
1.	4936	2341	219	534	49	113
3.	4.43×10^4	606	1027	2251	194	461
5.	7.03×10^{12}	2.74×10^7	132	130	115	115
8.	1.21×10^3	969	1148	1195	678	958
12.	1.08×10^6	5.55×10^3	3.29×10^3	7.3×10^3	264	373
14.	1.13×10^8	8.7×10^4	100	100	99	99
17.	4.16×10^8	4.62×10^5	793	790	713	714

1. F16 aircraft, 3. Binary distillation column, 5. Double inv. pendulum, 8. Quansar helicopter, 12. Tank reator 14. Two-link robot arm, 17. Spring-mass

sacrifices more sparsity for conditioning and yields a better-conditioned basis in general. However, Method 2 may encounter a significant increase in sparsity, e.g. in problems 5 and problem 14. With using restrictive criteria in column combination in Method 2A, such significant increases are prevented.

For the Quansar helicopter problem, the tuning methods 2 and 2A do not work well compared with the other problems in Table 3.2. One reason could be Method 2 and Method 2A seem to work well with well banded null bases. In the Quansar helicopter problem, the banded null basis (see Fig. 3.9) is “less structured”, therefore Method 2 and 2A work less effectively in this problem.

Table 3.3: Percentage increase of non-zero entries when trading sparsity for conditioning

Problem	Me.2	Me.2A
1. ATFI-F16	19.9	4.66
3. A binary distillation column	7.4	1.71
5. Double Inverted pendulum	19.9	4.66
8. Quansar helicopter	81.6	4.02
12. Tank Reactor	19.9	4.66
14. Two-link robot arm	42.9	8.15
17. Springmass system	12.85	2.8

Figure 3.10: Camera field-of-view quadrotor system [3]

3.7 Case study 2: embedded quadrotor application

The example is from [3]. In this example, a quadrotor is required to fly within the field-of-view of a camera, see Fig. 3.10. In [3], an interior-point-based MPC controller was implemented in MATLAB running in a ground-based PC. It was reported that the MPC controller runs at about 100Hz, however no detail is given on the specification of the PC hardware nor the MPC control horizon. The control command was then sent to the quadrotor via wireless communication.

The matrices in the system's state-space representation are

$$A = \text{diag}\{A_e, A_e, A_e\}, \quad B = \text{diag}\{B_e, B_e, B_e\}$$

Table 3.4: *Ave. runtime (ms)/ ave. iteration* for camera-field-of-view problem. The runtime is averaged over 100 random initial states. Notation: ADMM-B-RH : ADMM with the reduced Hessian and banded null basis; ##: CVXGEN (an interior-point QP algorithm of [4]) is unable to generate C-code for big problem.

	ADMM-B-RH	CVXGEN
N=5	2.5 /30	7.95 /5
N=10	7.0 /35	22.9 /5
N=15	11.1 /37	35.6 /5
N=20	17.9 /38	##
N=25	22.4 /40	##

Table 3.5: Code size (kB) (Notation: ADMM-B-RH : ADMM with the reduced Hessian and banded null basis; ##: CVXGEN unable to generate code for big dimension)

	ADMM-B-RH	CVXGEN
N=5	82	1552
N=10	102	3488
N=15	122	5314
N=20	143	##
N=25	163	##

where

$$A_e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & dT & 1 \\ 0 & 0 & (dT)^2/2 \end{bmatrix}, \quad B_e = \begin{bmatrix} 1 \\ 0 \\ (dT)^2/2 \end{bmatrix}, \quad \text{and the sampling time } dT = 0.01$$

The state and input dimension are $n_x = 9$, $n_u = 3$, and the polytopic constraints are $G_x x \leq f_x$ and $G_u u \leq g_u$ where $G_x \in \mathbb{R}^{6 \times 9}$, $G_u \in \mathbb{R}^{6 \times 3}$, i.e $n_{ix} = 6$, $n_{iu} = 6$, is obtained from the cited paper. The MPC parameters are $Q_x = I_{n_x}$, $R = 0.5I_{n_u}$, $Q_N = dare(A, B, Q_x, R)$ (*dare()*: MATLAB command for solving the discrete algebraic Riccati equation), $N = 5, 10, 15, 20, 25$.

In this example, Xilinx Software Development Kit (SDK) 14.6 is used for implementation on the ARM-Cortex A9 processor (667 MHz, single core, bare metal, double precision) embedded in the Zynq ZC 702 board. Since there are polytopic constraints in the MPC problem, only indirect ADMM and the banded reduced Hessian approach are applicable. As the banded reduced-Hessian outperforms indirect ADMM (see the author's earlier work [66]), only banded reduced Hessian is reported here.

A banded null basis Y is obtained and shown in Fig. 3.11. Table 3.4 show that ADMM to solve the banded reduced-Hessian QP (ADMM-B-RH) is about 3 times faster than

Figure 3.11: Sparsity pattern of F and its null space basis Y in the camera-field-of-view quadrotor problem.

CVXGEN, an interior-point QP solver package of [4], in this particular MPC problemⁱⁱⁱ.

Code size

Comparing code size shown in Table 3.5 shows that code size of ADMM-B-RH is small and grows slowly with respect to the control horizon. CVXGEN uses second order QP solver and usually generates a bigger code size compared with first order methods such as ADMM.

Computational growth versus control horizon

Fig. 3.12 shows the timing per iteration of ADMM to solve the banded Hessian QP. It shows that the time per iteration grows with a linear manner with respect to horizon N . This may be due to the fact that the sparsity of banded null basis only grows linearly with respect to the control horizon as described in the Conjecture 3.4.1.

This case study suggests that implementing an ADMM-based MPC for non-trivial problems on moderate microcontrollers is a viable task.

ⁱⁱⁱStopping criteria for ADMM: $\max\{\text{primal residual}, \text{dual residual}\} \leq 10^{-4}$. Time for optimality check is not counted. Number of iterations of CVXGEN is fixed at 5.

Figure 3.12: Time per ADMM iteration versus prediction horizon of reduced-Hessian-based ADMM using banded null basis

3.8 Automatic code generation for embedded implementation of MPC

In recent years, several automatic C-code generation tools for MPC controller have been developed, allowing the deployment of MPC controller be easier and time-saving. At the core of each tool is an optimisation method. qpOASES [67] uses an active-set method, CVXGEN [4] and FORCES [16] use interior-point-based methods, FiOrdOs [68] and μ AO-MPC [69] employ first order methods, QPgen [36] and OSQP [70] use splitting methods including ADMM.

A toolchain based on the proposed ADMM-based algorithm and banded null basis in this thesis also has been developed in MATLAB. This toolchain has been used for implementing an ADMM-based MPC controller in Simulink (for system verification before deploying), and one-click generating C code for embedded implementation on the EOR-3D device for a tap-changer transformer application. Fig 3.13 shows the closed-loop system. The first block (from the left) is the ADMM-based MPC controller which is configurable by user, the first block from the right is the plant. C-code generation is done by right-click on the MPC block and select build command. This toolchain is not far from to be a toolbox for automatic C code generation for MPC implementation, and may be publicly available in future.

Figure 3.13: MPC controller based on the proposed ADMM algorithms for a tap-changer transformer control problem

3.9 Conclusion

Techniques for exploiting the banded null basis that arises from MPC problem is presented. The proposed techniques were tested with ADMM algorithms. Results show that the time-per-iteration of direct ADMM and indirect ADMM were reduced significantly, compared with a standard implementation and the Schur reduction technique. The banded null basis is also used to obtain a reduced Hessian QP problem which has a small size as the condensed formulation; but since it still has banded matrices, it offers computational advantage. In addition, the proposed heuristics to trade off sparsity for conditioning work effectively, mitigating the problem with very ill-conditioned banded bases. For single-input system, an efficient algorithm for computing the banded basis is presented. In addition, the existence and sparsity bound for the banded null basis in MPC are proved. The banded null basis method can also be incorporated with interior-point methods, since equations similar to equation (3.1.1) must be solved.

3.10 Appendix

Appendix 3.10.1. The set of vectors

$$S_{\Omega} \triangleq \{B, AB, \dots, A^{\Omega-1}B\}$$

are linearly independent.

- (1) Assume that the number of columns E_k satisfying the condition of Step 4.1, i.e. active column condition, of Algorithm 3.4.2 is smaller than Ω . Therefore, the number of columns E_k not satisfying this condition is greater than $n_x - \Omega$. These columns (not satisfying 4.1)

are linearly independent and linearly independent with S_Ω . Hence, the total number of linearly independent columns within $S_\Omega \cup \{E_k\}_{k=1,2,\dots,n_x}$ is greater than $n_x (= \Omega + (n_x - \Omega))$. This cannot happen, since for any set of n_x -dimension vectors containing more than n_x vectors are linearly dependent [71].

(2) Assume that the number of columns E_k satisfying the condition of Step 4.1 of Algorithm 3.4.2 is bigger than Ω . Therefore, the number of columns E_k not satisfying this condition is smaller than $n_x - \Omega$. These columns (not satisfying 4.1) are linearly independent and linearly independent with S_Ω . Hence, the total number of linearly independent columns within $S_\Omega \cup \{E_k\}_{k=1,2,\dots,n_x}$ is smaller than $n_x (= \Omega + (n_x - \Omega))$. It follows that the dimension of the columns space of $\bar{S} = S_\Omega \cup \{E_k\}_{k=1,2,\dots,n_x}$ is smaller than n_x . This cannot happen since $\bar{S}$ contains n_x independent columns $\{E_k\}_{k=1,2,\dots,n_x}$.

The two above contradictions prove the lemma.

Appendix 3.10.2. For an active column E_k , it can be represented as linear combination of $E_{k-1}, \dots, E_1, B, AB, \dots, A^{\Omega-1}B$. Hence, there exist the smallest number $\Psi_k \leq \Omega - 1$ that E_k can be represented as the linear combination of $E_{k-1}, \dots, E_1, B, AB, \dots, A^{\Psi_k-1}B$.

Assume that there exist $i, j \in \{1, 2, \dots, n_x\}$ such that $\Psi_i = \Psi_j = \bar{\Psi}$ and $i \neq j$. Without lost of generality, assume that $i < j$.

From the definition of Ψ_i , it follows that $A^{\Psi_i}B$ is a linear combination of $E_i, E_{i-1}, \dots, E_1, B, AB, \dots, A^{\bar{\Psi}-1}B$. Since E_j is a linear combination of $E_{j-1}, \dots, E_i, E_{i-1}, \dots, E_1, B, \dots, A^{\Psi_j}B$, it then, by replacing A^{Ψ_j} by a linear combination of $\{E_i, E_{i-1}, \dots, E_1, B, AB, \dots, A^{\bar{\Psi}-1}B\}$, follows that E_j is a linear combination of $E_{j-1}, \dots, E_i, E_{i-1}, \dots, E_1, B, AB, \dots, A^{\bar{\Psi}-1}B$. This implies that $\Psi_j \leq \bar{\Psi} - 1$, i.e. $\Psi_j \leq \Psi_j - 1$. This contradiction proves the lemma. ■

Chapter 4

An ADMM algorithm for handling polytopic constraints and its fixed-point arithmetic implementation on a FPGA

Summary: In the previous chapter, banded null basis method has been presented to tackle the computational burden of the system of linear equations arising from ADMM-based MPC algorithms. This chapter presents method to tackle the challenges posed by the polytopic constraints and the step-size selection to embedded ADMM-based MPC applications. In addition, how to efficiently implement the algorithm on a Field Programmable Gate Array (FPGA), an emerging embedded platform, is also nontrivial. In this chapter, an algorithm for solving quadratic programming (QP) problems with inequality and equality constraints arising from linear MPC is proposed. The proposed algorithm is based on the ‘alternating direction method of multipliers’ (ADMM), with the introduction of slack variables. It can handle the so-called sparse MPC formulation with general inequality constraints. Moreover, the proposed algorithm is suitable for implementation on embedded platforms where computational resources are limited. In some cases, it is division-free when certain fixed matrices are computed offline. This enables it to be implemented in fixed-point arithmetic on a FPGA. Heuristic rules to select the step size of ADMM for a good convergence rate are also developed.

4.1 Algorithm description

When the state and control input constraints in MPC problem are polytopes, direct ADMM [44] will be computational demanding, since a projection operation onto the polytopes arises.

Slack variables technique has been employed in [37] to solve the condense QP. This section presents an ADMM algorithm for the sparse QP problem with polytopic constraints based on slack variable technique.

Slack variables s are added in the inequality constraint (2.1.9), i.e. $Gz + s = g$, the following problem is obtained:

$$\begin{aligned} \underset{z,s}{\text{minimize}} \quad & \frac{1}{2}z^T H z + I_{\mathbb{S}}(s) \end{aligned} \quad (4.1.1)$$

$$\text{s.t.} \quad \Psi z + \Gamma s = c \quad (4.1.2)$$

$$\mathbb{S} = \{s : s \geq 0\} \quad (4.1.3)$$

where

$$\Psi = \begin{bmatrix} G \\ F \end{bmatrix}, \quad \Gamma = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} g \\ f \end{bmatrix}, \quad (4.1.4)$$

$$I_{\mathbb{S}}(s) : \text{Indicator function of } \mathbb{S} \quad (4.1.5)$$

Applying the standard procedure of the ADMM method, ADMM-based algorithm for problem (4.1.1) is then as following

Algorithm 4.1 [72] (the author's earlier work)

Input: H, Ψ, c, g .

z_0, s_0, τ_0 **fixed** (Cold start)

Do

$$z_{k+1} = -(H + \rho \Psi^T \Psi)^{-1} [\rho \Psi^T (\Gamma s_k + \tau_k - c)] \quad (4.1.6)$$

$$\begin{aligned} s_{k+1} &= \pi_{\mathbb{S}}(-G z_{k+1} - \tau_k^g + g) \\ &= \max\{0, -G z_{k+1} - \tau_k^g + g\} \end{aligned} \quad (4.1.7)$$

$$\tau_{k+1} = \tau_k + \Psi z_{k+1} + \Gamma s_{k+1} - c \quad (4.1.8)$$

While $\|r_k\| \geq \epsilon_{\text{primal}}$ or $\|s_k\| \geq \epsilon_{\text{dual}}$

where $r_k = \Psi z_k + \Gamma s_k - c$, and $s_k = \rho \Psi^T \Gamma (s_{k+1} - s_k)$ are the primal and the dual residuals respectively; and the following Assumption is assumed to be satisfied

Assumption 4.1.1. $H + \rho\Psi^T\Psi$ is invertible.

Remark 4.1.1. $H_* = H + \rho\Psi^T\Psi = H + \rho F^T F + G^T G$. Then, H_* is invertible if one of following sufficient conditions is satisfied

- (i) H is positive definite (e.g. when all states and control inputs are penalized in the cost function).
- (ii) G is full column rank (then $G^T G$ will be positive definite). Explanation: for any z , if $z^T G^T G z = 0$ then $(Gz)^T (Gz) = 0$ implying $Gz = 0$. Since G is full column rank, $Gz = 0$ implies $z = 0$. Hence, $G^T G$ is positive definite implying that $H + \rho F^T F + G^T G$ is positive definite. One example when G is full column rank is all control inputs are imposed boxed constraints implying G contains an identity matrix.
- (iii) $Y^T H Y$ is positive definite where Y is a null basis of F . Explanation: for any z , if $z^T (H + \rho F^T F) z = 0$ then $z^T H z = 0$ and $z^T F^T F z = 0$, then $Fz = 0$ implying z belong to null space of F , i.e. there exists z_* such that $z = Yz_*$. It follows that $0 = z^T H z = z_*^T Y^T H Y z_*$. Since $Y^T H Y$ positive definite, it follows that $z_* = 0$. Therefore, $z = 0$. This implies that $H + \rho F^T F$ positive definite, hence $H + \rho F^T F + G^T G$ is positive definite

4.2 Step-size selection methods

The convergence rate of ADMM depends on the step size. To devise a step size selection method, [37] used the sign matrix of the variable technique of [73], to write the ADMM iterations in a matrix recurrence form. Later, in [74] [36] [75] different methods for step-size selection are presented.

In this section, methods for the selection of the step-size is derived for the proposed algorithm based on matrix recurrence form.

4.2.1 Matrix recurrence form

Let $\tau_k = \begin{bmatrix} \tau_k^g \\ \tau_k^f \\ \tau_k^g \end{bmatrix}$ where $\tau_k^g = \tau_k(1 : q)$ and $\tau_k^f = \tau_k(q + 1 : q + p)$ (q first elements and next p elements respectively, where $p = Nn_x$, $q = N(n_{ix} + n_{iu})$)

The ADMM iterations (4.1.6)-(4.1.8) can be rewritten as follows:

$$z_{k+1} = -(H + \rho\Psi^T\Psi)^{-1} \left[\rho\Psi^T \left(\begin{bmatrix} s_k \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_k^g \\ \tau_k^f \end{bmatrix} - c \right) \right] \quad (4.2.9)$$

$$s_{k+1} = \max\{0, -Gz_{k+1} - \tau_k^g + g\} \quad (4.2.10)$$

$$\begin{bmatrix} \tau_{k+1}^g \\ \tau_{k+1}^f \end{bmatrix} = \begin{bmatrix} \tau_k^g \\ \tau_k^f \end{bmatrix} + \begin{bmatrix} G \\ F \end{bmatrix} z_{k+1} + \begin{bmatrix} s_{k+1} \\ 0 \end{bmatrix} - \begin{bmatrix} g \\ f \end{bmatrix} \quad (4.2.11)$$

Introduce $v_k = z_k + \tau_k^g$.

From (4.2.10) and (4.2.11), it can be seen that $v_{k+1} = |Gz_{k+1} + \tau_k^g - g| = S_{k+1}(Gz_{k+1} + \tau_k^g - g)$, and $\tau_k^g = v_k - z_k = B_k v_k$ where

$$S_{k+1} = \text{diag}(\text{sign}(Gz_{k+1} + \tau_k^g - g)), \quad B_k = \frac{1}{2}(I + S_k) \quad (4.2.12)$$

Then, (4.2.9)-(4.2.11) become

$$z_{k+1} = -(H + \rho\Psi^T\Psi)^{-1} \left[\rho\Psi^T \left(\begin{bmatrix} v_k \\ \tau_k^f \end{bmatrix} - c \right) \right]$$

$$v_{k+1} = S_{k+1}(Gz_{k+1} + B_k v_k - g)$$

After some calculations, the following compact iterative formulation is obtained:

$$\begin{bmatrix} S_{k+1}v_{k+1} \\ \tau_{k+1}^f \end{bmatrix} = \frac{1}{2} \begin{bmatrix} S_k v_k \\ \tau_k^f \end{bmatrix} + \left(\frac{1}{2}I - \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T \right) \begin{bmatrix} v_k \\ \tau_k^f \end{bmatrix} + \begin{bmatrix} \bar{g} \\ \bar{f} \end{bmatrix} \quad (4.2.13)$$

where

$$\begin{aligned} \bar{g} &= \rho G(H + \rho\Psi^T\Psi)^{-1}\Psi^T c - g \\ \bar{f} &= \rho F(H + \rho\Psi^T\Psi)^{-1}\Psi^T c - f \end{aligned}$$

By defining the matrices as

$$\bar{S}_k = \begin{bmatrix} S_k & 0 \\ 0 & I_p \end{bmatrix}, \quad t_k = \begin{bmatrix} v_k \\ \tau_k^f \end{bmatrix}, \quad \gamma = \begin{bmatrix} \bar{g} \\ \bar{f} \end{bmatrix} \quad (4.2.14)$$

(4.2.13) is written as

$$\bar{S}_{k+1}t_{k+1} = \frac{1}{2}\bar{S}_k t_k + \left(\frac{1}{2}I - \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T \right) t_k + \gamma \quad (4.2.15)$$

Equation (4.2.15) is the matrix recurrence form of Algorithm 1. This is the same as equation (22) of [37]. However, the step-selection method proposed in [37] requires the

existence of H^{-1} , but in sparse QP formulation arising from MPC problems H is usually only semi-definite. Hence we propose heuristic rules to select the step-size for our proposed algorithm based on the following analysis.

4.2.2 Step-size selection

From (4.2.15) it follows that

$$\begin{aligned} \bar{S}_{k+1}t_{k+1} - \bar{S}_k t_k &= \frac{1}{2}(\bar{S}_k t_k - \bar{S}_{k-1} t_{k-1}) + \\ &\left(\frac{1}{2}I - \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T \right) (t_k - t_{k-1}) \end{aligned} \quad (4.2.16)$$

By taking norm ⁱ (which will be chosen later) of both sides of (4.2.16), and since $\|t_k - t_{k-1}\| \leq \|\bar{S}_k t_k - \bar{S}_{k-1} t_{k-1}\|$, it follows that:

$$\xi_{k+1} \leq \left(\frac{1}{2} + \left\| \frac{1}{2}I - M \right\| \right) \xi_k = \eta_\rho \xi_k \quad (4.2.17)$$

where

$\xi_k = \|(\bar{S}_k t_k - \bar{S}_{k-1} t_{k-1})\|$, $M = \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T$ and

$$\eta_\rho = \frac{1}{2} + \left\| \frac{1}{2}I - M \right\| = \frac{1}{2} + \left\| \frac{1}{2}I - \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T \right\| \quad (4.2.18)$$

Therefore, the optimal ρ can be obtained by minimising η_ρ over ρ . Different norms can be used, and in this paper, the following norms are investigated: (a) the 2-norm and (b) the max-norm to select the step-size parameter.

B.1. Step-size selection with respect to 2-norm

M is symmetric, since H is symmetric. As the 2-norm of a real symmetric matrix is the same as the spectral radius, the eigenvalues of the matrices appearing in the expression for η_ρ will be analysed.

Defining $T = \frac{1}{2}I - M$; T has eigenvalues in range $[-\frac{1}{2}, \frac{1}{2}]$ since M always has minimum eigenvalue in range $[0, 1]$ independent of ρ . If H is only (positive) semidefinite, M will have eigenvalues at 1, hence, $\eta_\rho = 1$ regardless of the choice of ρ . However, it is observed that if the other eigenvalues of T are shifted as close as possible to 0, or equivalently, move the eigenvalues of M as close as possible to $\frac{1}{2}$, the convergence speed becomes better.

Hence, as a heuristic, the following problem is solved to select ρ :

Problem 1: Find ρ such that the eigenvalues of $M = \rho\Psi(H + \rho\Psi^T\Psi)^{-1}\Psi^T$ are as close as possible to $\frac{1}{2}$.

In order to solve Problem 1, the eigenvalues of matrix M can be expressed explicitly as a

ⁱIn order for (4.2.17) to hold, we need the chosen norm to have sub-multiplicative property.

function of ρ . This can be achieved as follows: Since $H \succeq 0$, there exists $P \succeq 0$ such that $H = P^T P$ ([76] p8.3). Using the matrix inversion lemma [77], M can be written as

$$\begin{aligned} M &= \rho \Psi (H + \rho \Psi^T \Psi)^{-1} \Psi^T = \Psi (\rho^{-1} P^T P + \Psi^T \Psi)^{-1} \Psi^T \\ &= \Psi ((\Psi^T \Psi)^{-1} - \\ &(\Psi^T \Psi)^{-1} \rho^{-1} P^T (I + \rho^{-1} P (\Psi^T \Psi)^{-1} P^T)^{-1} P (\Psi^T \Psi)^{-1}) \Psi^T \end{aligned} \quad (4.2.19)$$

After some algebraic manipulations, M can be expressed as

$$\begin{aligned} M &= U (S S_d (I - \rho^{-1} P_d + \rho^{-2} P_d (I + \rho^{-1} P_d)^{-1} P_d) S_d^T S^T) U^T \\ &= U (S S_d \Psi_\rho S_d^T S^T) U^T \end{aligned} \quad (4.2.20)$$

where $\Psi = U S V^T$, the singular value decomposition of Ψ . Define $S_d = (S^T S)^{-\frac{1}{2}}$ ⁱⁱ, then

$$P_d = S_d^T V^T P^T P V S_d = S_d^T V^T H V S_d \quad (4.2.21)$$

$$\Psi_\rho = I - \rho^{-1} P_d + \rho^{-2} P_d (I + \rho^{-1} P_d)^{-1} P_d \quad (4.2.22)$$

Since $U^T = U^{-1}$, and the fact that the eigenvalues of two similar matrices are the same, it follows that $\lambda(M) = \lambda(S S_d \Psi_\rho S_d^T S^T) = \lambda(\bar{\Psi}_\rho)$, where $\bar{\Psi}_\rho = S S_d \Psi_\rho S_d^T S^T$.

As S is diagonal and singular, and S_d is diagonal, it can be shown that $\bar{\Psi}_\rho = \begin{bmatrix} \Psi_\rho & 0 \\ 0 & 0 \end{bmatrix}$.

Hence

$$\lambda(M) = \lambda(\bar{\Psi}_\rho) = \lambda(\Psi_\rho) \cup \{0\} \quad (4.2.23)$$

Then problem 1 is equivalent to the following problem:

Problem 2: Find ρ such that the elements of $\lambda(\Psi_\rho)$ are as near as possible to 0.5.

One way to solve Problem 2 is by solving:

$$\min_{\rho} \sum_{\lambda_i(\Psi_\rho) \neq 0} (\lambda_i(\Psi_\rho) - 0.5)^2 \quad (4.2.24)$$

Define

$$f(t) = 1 - t + t(1 + t)^{-1} t = \frac{1}{1 + t}$$

Based on [37], the eigenvalues of Ψ_ρ related to the eigenvalues of P_d as:

$$\lambda(\Psi_\rho) = f(\lambda(\rho^{-1} P_d)) = \frac{1}{1 + \rho^{-1} \lambda(P_d)} \quad (4.2.25)$$

ⁱⁱ S_d is diagonal matrix, since $S^T S$ is diagonal and positive definite (Assumption 1)

Hence, (4.2.24) is equivalent to:

$$\min_{\rho} \sum_{\lambda_i(P_d) \neq 0} \left(\frac{1}{1 + \rho^{-1} \lambda_i(P_d)} - 0.5 \right)^2 \quad (4.2.26)$$

It is not easy to solve (4.2.26). Since $\frac{1}{1 + \rho^{-1} \lambda_1(P_d)}$ and $\frac{1}{1 + \rho^{-1} \lambda_{max}(P_d)}$ are the maximum and minimum in the sequence $\frac{1}{1 + \rho^{-1} \lambda_i(P_d)}$, the following problem is solved instead:

$$\min_{\rho} \left(\frac{1}{1 + \rho^{-1} \lambda_1(P_d)} - \frac{1}{2} \right)^2 + \left(\frac{1}{1 + \rho^{-1} \lambda_{max}(P_d)} - \frac{1}{2} \right)^2 \quad (4.2.27)$$

Solving $\partial(.)/\partial\rho = 0$ leads to the solution of the following polynomial equation:

$$\begin{aligned} & \beta_1 \beta_2 (\beta_1^2 + \beta_2^2) \rho^4 - (\beta_1^3 + \beta_2^3 - 3\beta_1 \beta_2 (\beta_1 + \beta_2)) \rho^3 \\ & - (\beta_1 - \beta_2)^2 \rho^2 - 2(\beta_1 + \beta_2) \rho - 2 = 0 \end{aligned} \quad (4.2.28)$$

where $\beta_1 = (\lambda_1(P_d))^{-1}$, $\beta_2 = (\lambda_{max}(P_d))^{-1}$. In principle, one can obtain an analytical expression for the roots ρ of this 4th-degree equation. More practically, a numerical solution is obtained, and the positive root is selected. The procedure for determining ρ offline since Ψ and H do not change for MPC problems with linear time invariant models and fixed costs are then as the following:

1. Given Ψ and H
2. Calculate P_d as in (4.2.21), calculate non-zero eigenvalues $\lambda_1(P_d), \lambda_{max}(P_d)$, $\beta_1 = (\lambda_1(P_d))^{-1}$,
 $\beta_2 = (\lambda_{max}(P_d))^{-1}$
3. Solve (4.2.28) for ρ

B.2. Step-size selection with respect to max-norm

A simple line search is performed over ρ to minimise η_{ρ} directly, using the max-normⁱⁱⁱ. The advantage of using the max-norm is that it may handle general Ψ and H in the iterative form (4.2.17). This max-norm approach may be equally applicable to the method proposed in [37]. A more detailed analysis is the topic of current research. In Section 4.4.3, simulation studies suggest that the max-norm and the 2-norm criteria are both effective.

ⁱⁱⁱThe max-norm of a matrix is the largest absolute value of any element of the matrix. To be precise $(p+q) * \text{max-norm}$ should be used so that the sub-multiplicative property is satisfied. However, since $p+q$ is fixed, for simplicity, just say max-norm.

Figure 4.1: Basic FPGA architect consisting of configurable blocks

4.3 Implementation on an FPGA

Field Programmable Gate Arrays (FPGA) have emerged as popular platforms for real-time embedded MPC applications [27, 28, 78]. The simplicity of ADMM, and the parallel processing capability of FPGA, offer a promising combination for high speed embedded MPC. In this section, some implementation aspects of the proposed algorithm on an FPGA are discussed.

4.3.1 Offline and online computations

For MPC based on a linear time invariant model and with fixed costs and constraints, certain matrices can be computed offline, such as $M = -(H + \rho\Psi^T\Psi)^{-1}$ and $\bar{M} = \rho M\Psi^T$.

Variables which depend on measurements must of course be computed online, but the computations are relatively fast if matrices have been pre-computed. In particular, for the following algorithm $\bar{q} = Mq - \bar{M}c$ must be computed once each sampling time, when a new state measurement becomes available. Note also that $Bz = [z; 0]$.

Algorithm 4.2 Customized for embedded implementation

$$z_{k+1} = \bar{q} + \bar{M}([s_k; 0] + \tau_k) \quad (4.3.29)$$

$$s_{k+1} = \max\{0, -Gz_{k+1} - \tau_k^g + g\} \quad (4.3.30)$$

$$\tau_{k+1} = \tau_k + Az_{k+1} + [s_{k+1}; 0] - c \quad (4.3.31)$$

Remark 1: Equation (4.3.29) is one way to carry out the calculation step (4.3.29) which is the most computational step of Algorithm 4.2. This approach will result a matrix-vector product, hence favourable for efficient parallelism computation on FPGA. However, the sparsity of the matrices will be destroyed. An alternative option can be offline computation of the Cholesky factorisation of the banded matrix $Q + \rho\Psi^T\Psi$ and then (4.1.6) can be

carried out by back-substitution. This approach will reduce the computational cost and memory cost since the sparsity will be preserved.

4.3.2 Fixed-point arithmetic implementation

Since Algorithm 4.2 is division-free, it can be implemented in fixed-point arithmetic on a FPGA. Compared with floating point implementation, a fixed point implementation not only runs faster but also requires fewer hardware resources. In order to implement the algorithm using fixed-point arithmetic, the range of variables and the accuracy requirements need to be computed, in order to decide on the word length and number of fractional bits. An analytical bound of the variables are established as following: From equation (4.2.17), it follows that:

$$\begin{aligned}
\|t_k\|_2 &= \|\bar{S}_k t_k\|_2 = \|\bar{S}_0 t_0 + \sum_{j=1}^k (\bar{S}_j t_j - \bar{S}_{j-1} t_{j-1})\|_2 \\
&\leq \|\bar{S}_0 t_0\|_2 + \sum_{j=1}^k \|\bar{S}_j t_j - \bar{S}_{j-1} t_{j-1}\|_2 = \|\bar{S}_0 t_0\|_2 + \sum_{j=1}^k \xi_j \\
&\leq \|\bar{S}_0 t_0\|_2 + K_{max} \xi_1 = K_{max} \|\gamma\|_2
\end{aligned} \tag{4.3.32}$$

where γ is defined as equation (4.2.14), and K_{max} is the maximum number of iterations, a design choice.

The upper bound for $\|\gamma\|_2$ is known for a particular system. Then a bound for $\|t_k\|_2$ is obtained. A bound for $\|t_k\|_\infty$, i.e $\|t_k\|_\infty \leq \bar{t}$ also can be computed since $\|t_k\|_\infty \leq \|t_k\|_2$. Based on $\bar{t}$, the range of all variables as well as intermediate variables involved when implementing the algorithm can be established.

4.4 Case study

4.4.1 Illustration example

Figure 4.2: Spring-mass System

Consider the spring-mass system [79] shown in Fig. 4.2 with $M = 8$, , spring constant $k = 1 \text{ Nm}^{-1}$ and equal masses of $m = 1 \text{ kg}$ each. The states are the position and velocity of each mass, and the inputs are the forces on each mass. Hence the system has 16 states

and 8 inputs. The output $y = [y_1; y_2; \dots; y_8]$ contains the positions of the 8 masses. An MPC controller is designed with the following cost function and parameters:

$$\min \sum_{k=0}^{N-1} (x_k^T Q_x x_k + u_k^T R u_k) + x_N^T Q_N x_N \quad (4.4.33)$$

$$\text{s.t. } |y_i| \leq 4, |y_1 + y_2| \leq 6, |u_i| \leq 1 \quad (4.4.34)$$

The constraint $|y_1 + y_2| \leq 6$ is added to demonstrate the advantage of the proposed algorithm in handling non-box type constraints. In addition, to ensure Assumption 1 holds, a constraint on the velocities is set to a large value: $|\dot{y}_i| \leq 100$.

$Q_x = C_d^T C_d$, $R = I$ and terminal matrix Q_N is the solution of the discrete algebraic Riccati equation with parameters A_d, B_d, Q_x, R where A_d, B_d, C_d are the state-space matrices of the discrete-time system obtained by sampling the continuous system using a zero-order hold model with a sampling interval $T_s = 0.5$.

With horizon $N = 5$, the sparse QP formulation has 120 decision variables, 80 equality constraints, and 250 inequality constraints. Algorithm 1 was used to solve the QP problem online.

4.4.2 Result and discussion

Figure 4.3 shows the output and control on the fifth mass. The solution obtained by Algorithm 1 (red-dash square, implemented in MATLAB m-file), agreed with the solution obtained by calling the *quadprog* function in MATLAB (green line). Algorithm 4.1 is also implemented on the Zynq ZC702 FPGA^{iv} (blue-dot circle) in fixed-point arithmetic (32 bit word and 17 fractional bit (for some variables), and 25 bit word 17 fractional bit (for some variables)). The result produced by the FPGA implementation is almost the same, with relative errors of about 1.56% compared with the solution obtained on a PC. On average, it took about 80 ADMM iterations^{vi} (or 215 ms) to solve one QP on the FPGA^v, when the step-size $\rho = 2.0815$ was chosen based on the 2-norm method proposed in Section III.

- Figure 4.4(a) plots, as a function of ρ , the average number of iterations needed to solve QP problems arising from the first 10 sampling instants^{vi}.
- Figure 4.4(b) shows that the max-norm of matrix T (Section III.B.1) is a good indicator for the number of ADMM iterations.

^{iv}The hardware-in-the-loop is carried out, where the MPC controller runs in the FPGA board, and the spring-mass system is simulated in Simulink.

^{vi}Cold start, stopping criteria $\epsilon_{\text{primal}} = \epsilon_{\text{dual}} = 10^{-4}$ for each sampling instant

^vComputations are carried out sequentially. There is room for timing improvement if paralleling processing is employed, since Algorithm 1.1 mainly consists of dot-product computations.

Figure 4.3: MPC performance in Spring-Mass system

Although our heuristics did not give the optimal ρ , they nevertheless gave values that are close to the optimal ρ .

4.4.3 Comparison between 2-norm and max-norm

In Figure 4.5, additional QP problems with randomly selected MPC parameters are generated. The results showed that although different values of ρ were selected depending on the different norm used, the resulting number of iterations did not differ much for a wide range of ρ values. It appears that the max-norm could be a better indicator for selecting the step size parameter ρ , although experience needs to be gained with a greater variety of examples.

Figure 4.4: Selection of ρ based on the 2-norm or max-norm criteria. (MPC parameters: $Q_x = C_d^T C_d$, $R = I$)

Figure 4.5: Selection of ρ based on the 2-norm or max-norm criteria. (Random MPC parameters: $Q_x \succeq 0$, $R \succ 0$)

4.5 Conclusion

An ADMM-based algorithm for QP with inequality and equality constraints arising from MPC problems was proposed in this chapter. By introducing slack variables, the algorithm is greatly simplified because the projection is now onto a positive orthant rather than a general polytopic set. Hence the algorithm can handle general inequality constraints without the complexity of polytope projection. Heuristic methods based on 2-norm and max-norm criteria to select the step-size of the ADMM iteration for good convergence rates are also proposed. Simulations confirmed that the heuristics are effective and that the max-norm approach may be more successful. Since the algorithm has simple computational structure, as well as being division-free if some offline computations are performed, it allows an efficient implementation, with fixed-point arithmetic, on embedded platforms such as an FPGA.

At the time this work appeared (in July 2015), methods for ADMM to handle polytopic constraints in sparse MPC were not available. Later, several ADMM variants have been presented using slack variables (but using a different setups) can also handle the polytopic constraints, see e.g [43].

Chapter 5

Event-triggered anytime control

Summary: Chapters 3 and 4 tackle the computation burden of MPC on embedded system. Networked control is another non-traditional application of MPC which also poses non-traditional challenges. This chapter presents the event-triggered sequence-based anytime control (E-SAC), which is an effective method for dealing with the time-varying processor availability and lossy communication channels in networked and embedded control systems. The main idea of E-SAC is, when computing resources and measurements are available, to compute a sequence of tentative control inputs and store them in a buffer for potential future use. In this thesis, the E-SAC is extended to schemes which feature multiple control laws. For ease of exposition, the case of two control laws, namely a coarse and a fine control law, is presented in detail. Numerical simulations show that, in some circumstances, the proposed multi-control law schemes achieve better closed-loop performance with a lower channel utilization than the basic E-SAC algorithm. Since the current stability analysis framework for the E-SAC, the State-dependent Random-time Drift approach, becomes combinatoric and difficult to use when extending to the multi-control law E-SAC, we develop a general model and a new stability analysis for E-SAC based on Markov jump systems. Using the new stability analysis, stochastic stability conditions of existing E-SAC are also recovered as a special case.

Notation: $\mathbb{N} = \{1, 2, \dots\}$ represents natural numbers, $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$; $\mathbb{R}$ represents real numbers, $\mathbb{R}_{\geq 0} \triangleq [0, +\infty)$, $\mathbb{R}_{>0} \triangleq (0, +\infty)$; $\{x\}_{\mathcal{K}}$ stands for $\{x_k : k \in \mathcal{K}\}$, $\mathcal{K} \subseteq \mathbb{N}_0$. $\sigma(M)$ denotes the spectral radius of matrix M . $|x| = \sqrt{x^T x}$ denotes the Euclidean norm of vector x . A function $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}_\infty$ ($\phi \in \mathcal{K}_\infty$) if it is continuous, zero at the origin, strictly increasing and unbounded. $\mathbf{Pr}\{\Omega\}$, $\mathbf{Pr}\{\Omega|\Gamma\}$ denote the probability of an event Ω , and the conditional probability of Ω given Γ respectively. The expected value of a random variable ν given Γ is denoted by $\mathbf{E}\{\nu|\Gamma\}$, and $\mathbf{E}\{\nu\}$ represents the unconditional expectation. For a vector y , $y \succ 0$ means that all of its elements are positive. For a matrix

Figure 5.1: Setup of a system with an Event-triggered Sequence-based Anytime (E-SAC) algorithm.

$A \in \mathbb{R}^{n \times n}$, $A_{[i:j;l:m]}$ denotes a block matrix contained in A whose elements are taken from row i to j , and column l to m of A . For $z \in \mathbb{R}$, $\lfloor z \rfloor$ denotes the largest integer that is not bigger than z . For $n, k \in \mathbb{N}$, $n \pmod k$ means the remainder of n divided by k ; $\mathbf{0}_n$ is a zero vector with dimension n . For a vector x , $x(i)$ ($i \in \mathbb{N}$) denotes the i -th element of x . For a matrix A , $\|A\|_\infty$ denotes the infinity norm of A .

5.1 System model

Anytime control algorithm was first proposed in [49] to deal with time-varying processor availability. It uses an idea from the AI community called *anytime algorithm* [50], which is a computational procedure that could provide a valid answer even when it is terminated prematurely. In [49], different controllers with different floating point operations were designed. Notable later anytime control algorithms include [10, 51–53]. When sensor measurements are transmitted through a communication network, the measurements may be unavailable due to packet dropouts, or network congestion. Among anytime control algorithms, Sequence-based Anytime Control of [10] can handle this situation since it has a buffer which serves to provide a control input even when no measurement is received.

Motivated by the idea of using event-triggered control (see e.g. [54–58]) as a method to reduce demands on the network and computing processor while guaranteeing satisfactory levels of performance [59], SAC with an event-triggering mechanism (E-SAC) was proposed. Fig. 5.1 shows a system with the E-SAC algorithm.

In this system, the dynamics of the controlled plant is represented by:

$$x_{k+1} = f(x_k, u_k) \quad (5.1.1)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{U} \subseteq \mathbb{R}$, $k \in \mathbb{N}_0$, see Fig. 5.1.

Sensor measurements are transmitted to the controller via a delay-free communication link which introduces packet dropouts. The transmission is a threshold-based event-triggering, i.e., the sensor transmits the measurement only when $|x_k| > d$ where the threshold d is a design parameter. The threshold d is fixed once the system runs, and the triggering event is checked periodically at every sampling instant.

The outcome of the transmission is indicated by the random process $\{\gamma\}_{\mathbb{N}_0}$:

$$\gamma_k = \begin{cases} 0 & \text{if the sensor transmits but a dropout occurs}^i \\ 1 & \text{if } x_k \text{ is received successfully} \\ 2 & \text{if the sensor did not transmit (i.e. } |x_k| \leq d) \end{cases}$$

which is assumed to be (conditionally) independent and identical (i.i.d) with a successful transmission probability

$$q \triangleq \mathbf{Pr}\{\gamma_k = 1 | |x_k| > d\} = \mathbf{Pr}\{\gamma_k = 1 | \gamma_k \neq 2\} \quad (5.1.2)$$

Assumption 5.1.1 (Processor availability). The processor is triggered by arrival of valid data. The processor availability for control at different time-instants is (conditionally) i.i.d. Thus, we denote by $N_k \in \{0, 1, 2, \dots, N_{\max}\}$, how many processing units are available at time instants k . The process $\{N\}_{\mathbb{N}_0}$ has conditional probability distribution:

$$\mathbf{Pr}\{N_k = j | \gamma_k = 1\} = p_j, \quad j \in \{0, 1, 2, \dots, N_{\max}\}, \quad (5.1.3)$$

where $p_j \in [0, 1)$ are given.

For other values of γ_k , no plant input is calculated. Thus the processing resources are considered not available regardless, i.e.:

$$\mathbf{Pr}\{N_k = 0 | \gamma_k \in \{0, 2\}\} = 1$$

Remark 5.1.1. Processing units (in Assumption 5.1.1) represent the computational resources (e.g. memory units and given CPU time) available for computing the sequence of predicted control inputs. We assume that the processing time of the control task is significantly smaller than the sampling time of the plant model.

Assumption 5.1.2 (Coarse and fine control policy).

The coarse control law $\kappa_1 : \mathbb{R}^n \rightarrow \mathbb{U}$ requires 1 processing unit to compute, whereas the

ⁱIt is assumed that packet dropout ($\gamma_k = 0$) is distinguishable from no transmission ($\gamma_k = 2$), e.g., through error-detection coding and monitoring of received energy/waveforms in the sensor transmission band (see [80]).

fine control law $\kappa_2 : \mathbb{R}^n \rightarrow \mathbb{U}$ requires $\eta \in \mathbb{N}$, $\eta \geq 1$, processing units to compute. It is assumed that there exist a common Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$; $\varphi_1, \varphi_2 \in \mathcal{K}_{\infty}$, and $\rho_1 \in \mathbb{R}_{\geq 0}$, $\alpha > 0$, such that $\forall x \in \mathbb{R}^n$

$$\varphi_1(|x|) \leq V(x) \leq \varphi_2(|x|) \quad (5.1.4)$$

$$V(f(x, 0)) \leq \alpha V(x) \quad (\text{open-loop bound}) \quad (5.1.5)$$

$$V(f(x, \kappa_1(x))) \leq \rho_1 V(x) \quad (\text{closed-loop contraction 1}) \quad (5.1.6)$$

$$V(f(x, \kappa_2(x))) \leq \rho_2 V(x) \quad (\text{closed-loop contraction 2}) \quad (5.1.7)$$

and the fine control policy κ_2 is better than the coarse policy κ_1 in the sense that

$$\rho_2 < \rho_1$$

■

Remark 5.1.2. Assumption 5.1.2 is potentially conservative as a common Lyapunov function is required. It limits the class of system and controller that are applicable under the presented framework in this thesis. This assumption is a combination of some Assumptions from [10] and [19]. As showed in two examples of [10], the selection the Lyapunov function and the control laws is problem-dependent. Since the scope of this chapter is stability analysis, the design of the control laws and Lyapunov function is not considered. It is a future research topic. However, there exist some textbooks discussing the design of control laws and Lypunov function for some classes of non-linear systems, e.g., see [81].

The baseline algorithm, here denoted by $B1$, amounts to a direct implementation of κ_1 as per

$$u_k = \begin{cases} \kappa_1(x_k) & \text{if } \gamma_k = 1 \text{ and processor is available} \\ 0 & \text{otherwise} \end{cases}$$

Fig. 5.2 shows the operation of the (one-control law) E-SAC in [11], denoted as $A1$. In $A1$, tentative future inputs using κ_1 are calculated and stored in a local buffer $b_k \in \mathbb{R}^{\Lambda}$ (Λ : buffer size, the maximum number of control inputs it can store), whenever the computing resources are available ($\gamma_k = 1$ & $N_k > 0$). When $|x_k| > d$ and processing resources are unavailable (due to dropouts or unavailable processor, i.e. $\gamma_k = 0$ or $N_k = 0$), the buffer is shifted, i.e., the first element is thrown away and the rest is kept. In case the system state lies within the threshold, i.e., $|x_k| \leq d$ or $\gamma_k = 2$, the buffer is cleared. The matrix

Figure 5.2: Operation of A1, b_k : buffer content at time instant k .

representing the shift operation is defined as

$$\Xi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{\Lambda \times \Lambda}$$

After completing the filling of the buffer, the first element in the buffer is used for the current input. A1 is referred to as *one-control law* E-SAC scheme. For more details, see [11].

5.2 E-SAC with multiple control laws

In control system design, at times one may encounter situations where one would like to switch between different control laws in respond to changing operating conditions. For example, depending on available computational resources, one may switch between a suboptimal or optimal controller, a short or long prediction horizon MPC, or a fixed- or floating-point controller implementation. In [19], E-SAC was extended to schemes featuring two control laws, κ_1 and κ_2 , to capture such situations. The control laws κ_1 and κ_2 are refereed to as the coarse (baseline) control law and the fine control law respectively. The fine control law κ_2 requires more computational resource to execute than κ_1 as shown

in Assumption 5.1.2.

Algorithm *B2* and *A2* below are described in a condense version compared with them presented in [19].

5.2.1 Algorithm *B2*: two-control law E-SAC without buffer

Algorithm *B2* amounts to a direct implementation of κ_1 and κ_2 without any buffering. The plant input is calculated as

$$u_k = \begin{cases} \kappa_2(x_k), & \text{if } N_k \geq \eta \\ \kappa_1(x_k), & \text{if } 0 < N_k < \eta \\ 0, & \text{if } N_k = 0. \end{cases}$$

5.2.2 Algorithm *A2*: two-control law E-SAC with buffer

Fig. 5.3 shows the operation of *A2*. Similar to *A1*, a local buffer with contents $b_k^\dagger$ of size $\Lambda \in \mathbb{N}$ is used to store the sequence of tentative future plant inputs calculated by either κ_1 or κ_2 at time k using excess processing resources.

To be more specific, the control policies κ_1 or κ_2 and their tentative future sequence will be executed depending on the values of N_k , as illustrated next.

Given the available processing unit $N_k > 0$ (assumed known in advance), we can write

$$N_k = \tau_k \eta + M_k$$

where $\tau_k \triangleq \lfloor \frac{N_k}{\eta} \rfloor$, and $M_k = N_k \bmod \eta$ (η : see Assumption 5.1.2). Firstly, a tentative control sequence is computed by iterating τ_k times the model (5.1.1) using κ_2 , and then by iterating M_k times the model (5.1.1) using κ_1 . When $|x_k| > d$ and computing resources are unavailable (due to dropouts or unavailable processor), the buffer is shifted. In case the norm of the system state lies within the threshold, i.e., $|x_k| \leq d$ ($\Leftrightarrow \gamma_k = 2$), the buffer is cleared. The first element in the buffer will be used as the current input. We refer to *A2* as *two-control law E-SAC* scheme. For ease of exposition, in this work, we only present in details the case of two control laws. The case of multiple control laws can be adapted directly.

Remark 5.2.1. It is important to note that the control values written into the buffer at time k only use information about x_k (if available), but not x_{k+1} , or other future states. Merely predictions are used. Therefore, and assuming that processing and transmissions are “infinitely” fast, it is appropriate to use a time-invariant system model as (5.1.1). The

Figure 5.3: Operation of A2, $b_k^\dagger$: buffer content at time instant k .

overall system (including communications, and computations) turns out to be stochastically switching, leading to non-trivial dynamics.

Example 5.2.1. Suppose that $N_{\max} = 3$, $\eta = 2$, $\Lambda = 3$ and that the processor availability is such that $N_0 = 3$, $N_1 = 0$, $N_2 = 2$; the system state is such that $|x_k| > d$, $\forall k = 0, 1, 2$ and there are no dropouts.

If algorithm A1 is used with κ_1 , then the buffer contents become:

$$\{b_0, b_1, b_2\} = \left\{ \begin{bmatrix} \kappa_1(x_0) \\ \kappa_1(f(x_0, \kappa_1(x_0))) \\ \kappa_1(\hat{x}_0^{[\kappa_1, 2]}) \end{bmatrix} \text{ ii, } \begin{bmatrix} \kappa_1(f(x_0, \kappa_1(x_0))) \\ \kappa_1(\hat{x}_0^{[\kappa_1, 2]}) \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_1(x_2) \\ \kappa_1(f(x_1, \kappa_1(x_2))) \\ 0 \end{bmatrix} \right\}$$

which gives the plant inputs $u_0 = \kappa_1(x_0)$, $u_1 = \kappa_1(f(x_0, \kappa_1(x_0)))$, $u_2 = \kappa_1(x_2)$.

ii $\hat{x}_0^{[\kappa_1, 2]} = f(f(x_0, \kappa_1(x_0)), \kappa_1(f(x_0, \kappa_1(x_0))))$

If algorithm *A2* is used, then the buffer contents at times $k \in \{0, 1, 2\}$ become:

$$\{b_0^\dagger, b_1^\dagger, b_2^\dagger\} = \left\{ \begin{bmatrix} \kappa_2(x_0) \\ \kappa_1(f(x_0, \kappa_2(x_0))) \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_1(f(x_0, \kappa_2(x_0))) \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_2(x_2) \\ 0 \\ 0 \end{bmatrix} \right\}$$

which gives the plant inputs $u_0 = \kappa_2(x_0)$, $u_1 = \kappa_1(f(x_0, \kappa_2(x_0)))$, $u_2 = \kappa_2(x_2)$.

For the no-buffering schemes, the plant inputs are $u_0 = \kappa_1(x_0)$, $u_1 = 0$, $u_2 = \kappa_1(x_2)$ for algorithm *B1*, and $u_0 = \kappa_2(x_0)$, $u_1 = 0$, $u_2 = \kappa_2(x_2)$ for algorithm *B2*.

This example suggests that *A2* outperforms *A1* since κ_2 gives better control inputs than κ_1 . As illustrated, the no-buffering schemes *B1* and *B2* cannot provide a control input when the processor is unavailable at time step $k = 1$.

5.2.3 State-dependent random-time drift condition approach for stability analysis of E-SAC

In [11], the state-dependent random-time drift (SRD) condition is developed to derive the stochastic stability of the one-control law scheme with buffering, i.e. the scheme *A1*. For deriving the stability condition, it requires one to calculate probability mass function (pmf) of random variable Δ_i , which denotes the time interval that the buffer becomes empty again. An analytical formulation of this pmf as well as the closed-form for stability boundary of *A1* has been established in [11].

In [19], the derivation of stability condition for the two-control law with buffering, i.e. scheme *A2*, follows the same SRD method as above. Since there are two control laws in the buffer, the fine control law κ_2 and the coarse control law κ_1 , there are not only Δ_i is a random variable, but also the number of times each control law is active, denoted by r_i , is also a random variable, see Fig. 5.4. Therefore, it is a combinatoric problem and quickly becomes intractable. Details of SRD for *A2* can be found in the Appendix 5.7.7.

5.3 Markov Jump System model and stability framework

This section proposes a different approach for the stability analysis of E-SAC based on Markov jump system ideas.

A stochastic model of the buffer contents at any time k is developed first.

Figure 5.4: Illustration of the State-dependent Random-time Drift condition approach in the E-SAC with two control laws. $\{k_0, k_1, \dots\}$: time steps that the buffer is empty, $\Delta_i = k_{i+1} - k_i$ and r_i (number of times the coarse control law is active during $[k_i, k_{i+1}]$) are random variables.

5.3.1 Markov state of the buffer content

For the one-control law E-SAC, during periods when $|x_k| > d$, the buffer length is a Markov chain since it only depends on N_k which is an i.i.d random process. We model the buffer length as the random process $\{\theta_k\}$ with the corresponding state space:

$$\mathcal{S}^{(A1)} = \{0, 1, 2, \dots, N_{\max}\} \triangleq \{S_i\}_{i=1, \dots, N_{\max}+1} \quad (5.3.8)$$

and the conditional ($|x_k| > d$) transition probability matrix

$$\Pi^{(A1)} = \{\pi_{ij}^{(A1)}\}, \quad i, j = 1, 2, \dots, N_{\max} + 1$$

where $\pi_{ij}^{(A1)} = \mathbf{Pr}\{\theta_k = S_j | \theta_{k-1} = S_i, \gamma_k \neq 2, \gamma_{k-1} \neq 2\}$ is shown in the Appendix A, and

$$\Pi^{(A1)} = \begin{bmatrix} l_0 & l_1 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ l_0 & l_1 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ 0 & l_1 + l_0 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ 0 & l_1 & l_2 + l_0 & l_3 & \cdots & l_{N_{\max}} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & l_1 & \cdots & \cdots & l_{N_{\max}-1} + l_0 & l_{N_{\max}} \end{bmatrix} \quad (5.3.9)$$

Figure 5.5: State transition diagram for buffer length of A1 ($N_{max} = 4$), when $|x_k| > d$

In (5.3.9),

$$\begin{aligned}
 l_j &\triangleq \mathbf{Pr}\{N_k = j | \gamma_k \neq 2, \gamma_{k-1} \neq 2\} = \mathbf{Pr}\{N_k = j | \gamma_k \neq 2\} \\
 &= \mathbf{Pr}\{N_k = j | \gamma_k = 1\} \mathbf{Pr}\{\gamma_k = 1 | \gamma_k \neq 2\} \\
 &\quad + \mathbf{Pr}\{N_k = j | \gamma_k = 0\} \mathbf{Pr}\{\gamma_k = 0 | \gamma_k \neq 2\}
 \end{aligned} \tag{5.3.10}$$

satisfy $l_0 = p_0 q + (1 - q)$, $l_j = p_j q$ ($j = 1, 2, \dots, N_{max}$), where p_j and q are defined as in (5.1.2) and (5.1.3).

As an example, if $N_{max} = 4$, we have the state space: $\mathcal{S}^{(A1)} = \{S_i\}_{i=1, \dots, 5} = \{0, 1, 2, 3, 4\}$ and the transition diagram (when $|x_k| > d$) is shown in Fig. 5.5.

For two-control law scheme A2, the content of the buffer is modelled via $\theta_k = (F_k; C_k)$ where F_k and C_k indicate the number of κ_2 (fine control law) and κ_1 (baseline or coarse control law) in the buffer at time step k respectively. During periods when $|x_k| > d$, then the transition of θ_k only depends on N_k which is i.i.d. Hence, during periods when $|x_k| > d$, $\{\theta_k\}$ is a Markov chain. The corresponding state space is

$$\begin{aligned}
 \mathcal{S}^{(A2)} &= \{S_i = (\lfloor \frac{i-1}{\eta} \rfloor; (i-1) \bmod \eta)\}, \\
 i &= 1, 2, \dots, N_{max} + 1
 \end{aligned} \tag{5.3.11}$$

and is associated with the conditional ($|x_k| > d$) transition probability matrix

$$\Pi^{(A2)} = \{\pi_{ij}^{(A2)}\}, \quad i, j = 1, 2, \dots, N_{max} + 1.$$

5.3.2 Markov jump system with event triggering model

For our subsequent analysis, it is convenient to introduce

$$y_k \triangleq [x_k; x_{k-1}; \dots; x_{k-\Lambda+1}] \in \mathbb{R}^{n\Lambda} \quad (5.3.12)$$

Lemma 5.3.1. $\{y\}_{\mathbb{N}_0}$ is Markovian.

Proof. See Appendix 5.7.2. □

From (5.1.1) and (5.3.12), for the process $\{y_k\}$, there exists $\psi : \mathbb{R}^{n\Lambda} \times \mathbb{U} \rightarrow \mathbb{R}^{n\Lambda}$ such that $y_{k+1} = \psi(y_k, u_k)$.

The control u_k is determined by y_k and the random process θ_k describing the buffer contents (see Remark 5.3.2). Then, in general, we have $u_k = \hat{\kappa}_{(\theta_k)}(y_k)$ where $\theta_k \in \{S_i\}_{i=1,2,\dots}$ (see (5.3.11) for S_i) and set of control laws, associated with the contents of the buffer, $\hat{\kappa}_{(S_i)} : \mathbb{R}^{n\Lambda} \rightarrow \mathbb{U}$. This leads to

$$y_{k+1} = \psi(y_k, \hat{\kappa}_{(\theta_k)}(y_k)) \quad (5.3.13)$$

which is a Markov jump system during intervals when $|x_k| > d$. Here, $\hat{\kappa}$ presents the mappings from the domain of y_k (which is different from the domain of x_k) to the domain of u_k ⁱⁱⁱ.

The mapping from a Markov state to a control law for $\{y_k\}$ process is

$$\begin{aligned} \hat{\kappa}_{(\theta_k=S_0)}(y_k) &= 0, \\ \hat{\kappa}_{(\theta_k=S_1)}(y_k) &= \kappa_1(x_k), \quad \dots, \quad \hat{\kappa}_{(\theta_k=S_\eta)}(y_k) = \kappa_1(x_k), \\ \hat{\kappa}_{(\theta_k=S_{\eta+1})}(y_k) &= \kappa_2(x_k), \quad \dots, \quad \hat{\kappa}_{(\theta_k=S_{\Lambda+1})}(y_k) = \kappa_2(x_k) \end{aligned} \quad (5.3.14)$$

Figure. 5.7 shows an equivalent model of the E-SAC schemes A2 via the process $\{y_k\}$. This model is defined as the event-triggered Markov jump system model (E-MJS). $\beta_k \in \{1, 0\}$ represents the threshold-based triggering event $|x_k| > d$ ($\gamma_k \in \{0, 1\}$) and $|x_k| \leq d$ ($\gamma_k = 2$). In this model, the particular schemes such as A1 and A2 are encoded in the state space $\{S_1, S_2, \dots\}$ (for e.g. (5.3.11)). The dropouts and processor availability are encoded in the transition probabilities of the state space.

ⁱⁱⁱ κ_1, κ_2 are the mapping from the domain of x_k to the domain of u_k .

Figure 5.7: E-MJS model of E-SAC schemes.

5.3.3 General model and stability analysis framework

A general mathematical description for the E-MJS model of the E-SAC is proposed as follows. Consider a non-linear system $y_{k+1} = \psi(y_k, u_k)$ controlled by two controllers: (1) stochastic controller and (2) deterministic controller. The loop is closed with either the stochastic or the deterministic controller (Fig. 5.7). When the triggering condition is met, the stochastic controller will be deployed. We use $\beta_k \in \{0, 1\}$ to indicate the triggering event at time k :

$$\beta_k = \begin{cases} 0 & \text{use deterministic controller} \\ 1 & \text{use stochastic controller} \end{cases}$$

Stochastic controller

Due to the external environment, such as time-varying processing powers or dropouts in the communication channels, the controller switches stochastically within a set of M control laws $\{\hat{\kappa}_{(i)}, i = 1, 2, \dots, M\}$.

In this case, the closed loop system model is

$$y_{k+1} = \psi(y_k, u_k) = \psi(y_k, \hat{\kappa}_{(\theta_k)}(y_k))$$

where $\{\theta_k\}_{\mathbb{N}}$ is a discrete Markov chain with state space

$$\mathcal{S} = \{1, 2, \dots, M\} \quad (5.3.15)$$

and the (conditional) transition probability matrix

$$\Pi = \{\pi_{ij}\}, \quad i, j = 1, \dots, M$$

where $\pi_{ij} = \mathbf{Pr}\{\theta_{k+1} = j | \theta_k = i, \beta_k = 1, \beta_{k+1} = 1\}$

Here, for ease of notation, numeric representation is used for representing the state space of the Markov chain in (5.3.15) instead of using S_i as (5.3.11).

Deterministic controller

This controller gives a fixed control policy $u_k = \hat{\kappa}_{(-1)}(y_k)$ and in this case, the closed loop is $y_{k+1} = \psi(y_k, \hat{\kappa}_{(-1)}(y_k))$.

For simplicity but without loss of generality, $\kappa_{(-1)}(y_k) = 0$.

5.3.4 Stochastic stability of E-MJS model

In this section, the stochastic stability condition for the proposed E-MJS model are derived.

First, the following assumptions are made:

Assumption 5.3.1. There exists a non-negative function $\tilde{V} : \mathbb{R}^{n_y} \rightarrow \mathbb{R}_{\geq 0}$ (n_y is dimension of y) and coefficients $\sigma_{(i)} \in \mathbb{R}_{\geq 0}$, $i = -1, 1, 2, \dots, M$ such that

$$\tilde{V}(\psi(y, \hat{\kappa}_{(i)}(y))) \leq \sigma_{(i)} \tilde{V}(y), \quad \forall i = -1, 1, 2, \dots, M \quad (5.3.16)$$

Assumption 5.3.2. There exists a constant $D \in \mathbb{R}_+$ such that $\tilde{V}(y_k) \leq D$, if the deterministic controller setup is in operation. ■

Remark 5.3.3. Assumption 5.3.1 characterises each control law $\hat{\kappa}_{(i)}$ by a scalar $\sigma_{(i)}$, and bounds the rate of increase of $\tilde{V}(y)$ when a control law $\hat{\kappa}_{(i)}$ is active. In Section 5.4.1 we show that Assumptions 5.3.1 and 5.3.2 are satisfied in the E-SAC schemes A1 and A2, whenever Assumptions 5.1.2 is satisfied. However, Assumptions 5.1.2 is potentially conservative as a common Lyapunov function is required. ■

Let $\tilde{V}_k = \tilde{V}(y_k)$, then the following is the stochastic model for $\{\tilde{V}_k\}_{\mathbb{N}_0}$:

$$\begin{cases} \tilde{V}_{k+1} \leq \sigma_{(\theta_k)} \tilde{V}_k, & \underbrace{(\theta_k \in \{1, 2, \dots, M\})}_{\text{Markov jump}} \quad \text{if } \beta_k = 1 \\ \tilde{V}_{k+1} \leq \sigma_{(-1)} \tilde{V}_k & \text{if } \beta_k = 0 \end{cases}$$

or in a compact form as:

$$\tilde{V}_{k+1} \leq \sigma_{(\theta_k)} \tilde{V}_k, \quad (\theta_k \in \underbrace{\{-1\}}_{\text{(deterministic ctrl.)}} \cup \underbrace{\{1, 2, \dots, M\}}_{\text{Markov jump, stochastic ctrl.}}) \quad (5.3.17)$$

Note that the range of θ_k is extended to include $\theta_k = -1$ if $\beta_k = 0$ (deterministic mode) to have the compact form as shown in (5.3.17).

Theorem 5.3.1. If there exist positive real numbers $\nu_{(1)}, \dots, \nu_{(M)}$, and $\zeta_{(1)}, \dots, \zeta_{(M)}$ such that

$$\sum_{j=1}^M \pi_{ij} \sigma_{(i)} \zeta_{(j)} - \zeta_{(i)} = -\nu_{(i)}, \quad (5.3.18)$$

for all $i = 1, \dots, M$, then $\mathbf{E}\{\tilde{V}_k\} < C_1 \xi^k \mathbf{E}\{\tilde{V}_0\} + C_2 < \infty$ where

$$\xi = 1 - \frac{\min_{1 \leq j \leq M} \{\nu_{(j)}\}}{\max_{1 \leq k \leq M} \{\zeta_{(k)}\}} \in (0, 1)$$

$$C_1 = \frac{\zeta_{\min}}{\zeta_{\min}}, \quad C_2 = \frac{1}{(\zeta_{\min})(1-\xi)} (\max\{\zeta_{\min} D, |\zeta_{\max} \sigma_{(-1)} - \xi \zeta_{\min}| D\}) \quad (\zeta_{\max} = \max\{\zeta_{(i)}\}_{i=1, \dots, M}, \zeta_{\min} = \min\{\zeta_{(i)}\}_{i=1, \dots, M}).$$

Proof. The proof is essentially an adaptation of the general stability result of Markov jump linear systems from [82], [83] specialised for (5.3.17), which is a scalar and positive system with event-triggering. See details in Appendix 5.7.3. $\square$

Theorem 5.3.1 provides a general condition for stochastic stability of $\{\tilde{V}_k\}$ in terms of the boundedness property of the expectation. Note that (5.3.18) represents a system of linear equations and can be represented as

$$(I - \Phi \Pi) \underline{\zeta} = \underline{\nu} \quad (5.3.19)$$

where

$$\Phi \triangleq \text{diag}\{\sigma_{(1)}, \sigma_{(2)}, \dots, \sigma_{(M)}\} \quad (5.3.20)$$

and $\underline{\zeta} \triangleq (\zeta_{(1)} \ \zeta_{(2)} \ \dots \ \zeta_{(M)})^T$, $\underline{\nu} \triangleq (\nu_{(1)} \ \nu_{(2)} \ \dots \ \nu_{(M)})^T$.

Then, Theorem 5.3.1 can be restated as follows:

Corollary 5.3.1. Define the certification matrix $\mathcal{T} = \Phi\Pi$. If $\mathcal{T}$ is Schur stable, then $\mathbf{E}\{\tilde{V}_k\} < C_1\xi^k\mathbf{E}\{\tilde{V}_0\} + C_2 < \infty$ where $\xi \in (0, 1)$, $C_1, C_2 \in \mathbb{R}_{>0}$.

Proof: See Appendix 5.7.4.

5.4 Stochastic Stability of E-SAC Schemes

In this section, stochastic stability conditions for the E-SAC schemes are derived by applying the results of Section 5.3.4.

5.4.1 Existence and bounds of $\tilde{V}(y_k)$ of E-SAC

For the process $\{y_k\}$ describing E-SAC (see (5.3.12)), the following function is chosen

$$\tilde{V} : \mathbb{R}^{n\Lambda} \rightarrow \mathbb{R}_{\geq 0}, \quad \tilde{V}(y_k) = V([I \ 0 \ 0 \ \dots \ 0]y_k) = V(x_k)$$

where $V(\cdot)$ is defined as in Assumption 5.1.2.

The reason for choosing this $\tilde{V}$ is that the bound $\sigma_{(i)}$ in (5.3.16) exists. This bound is related to a control law $\hat{\kappa}_{(i)}$ which is associated with a Markov state.

Assumptions 5.1.2 and equations (5.3.14), lead to the following bounds for $\tilde{V}(y_k)$

$$A2 : \begin{cases} \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_1)}(y_k))) \leq \alpha \tilde{V}(y_k) \\ \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_i)}(y_k))) \leq \rho_1 \tilde{V}(y_k), \quad i = 2, 3, \dots, \eta \\ \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_i)}(y_k))) \leq \rho_2 \tilde{V}(y_k), \\ \quad i = \eta + 1, \dots, N_{max} + 1 \end{cases} \quad (5.4.21)$$

where α , ρ_1 and ρ_2 are defined as in Assumptions 5.1.2.

To show that Assumption 5.3.2 is also satisfied, recall that in the deterministic controller setup, $|x_k| \leq d$. Therefore $\tilde{V}(y_k) = V(x_k) \leq \varphi_2(|x_k|) \leq \varphi_2(d) \triangleq D$ (see Assumption 5.1.2). ■

5.4.2 Stochastic stability for E-SAC schemes A1 and A2

Following Lemma is needed to establish closed-loop stability, when A1 or A2 are used.

Lemma 5.4.1. Consider a 2×2 block matrix $H = \begin{bmatrix} X & Y \\ Z & M \end{bmatrix}$, where $X \in \mathbb{R}^{1 \times 1}, Y \in \mathbb{R}^{1 \times m}, Z \in \mathbb{R}^{m \times 1}, M \in \mathbb{R}^{m \times m}$, and M is Schur stable with non-negative entries and $\|M\|_\infty < 1$ and $\text{trace}(M^2) < 1$. Then H is Schur stable if and only if $g(1) > 0$ where $g(\lambda) = (\lambda I_n - X) - Y(\lambda I_m - M)^{-1}Z$.

Proof: See Appendix 5.7.5. ■

Closed-loop stability when using algorithm A2 is then established as follows:

Corollary 5.4.1 (Stochastic stability of A2).

The E-SAC scheme A2 yields a stochastically stable loop, in the sense that $V(x_k)$ satisfies the bound condition in Theorem 5.3.1, if the certification matrix

$$\mathcal{T}^{(A2)} \triangleq \text{diag}\{\alpha, \underbrace{\rho_1, \dots, \rho_1}_{\eta-1 \text{ times}}, \underbrace{\rho_2, \dots, \rho_2}_{N_{max}-\eta+1 \text{ times}}\} \Pi^{(A2)}$$

is Schur stable.

Further, if $\rho_1, \rho_2 < 1$ the Schur stability of the certification matrix $\mathcal{T}^{(A2)}$ reduces to

$$\Psi \triangleq l_0 \alpha + l_0 \alpha \Theta_2^T (I - G_{\rho_1, \rho_2})^{-1} E_2 < 1 \quad (5.4.22)$$

where $E_2^T = [\rho_1 \quad \underbrace{0 \dots 0}_{\eta-2 \text{ elements}} \quad \rho_2 \quad \underbrace{0 \dots 0}_{N_{max}-\eta+1 \text{ elements}}]$, $\Theta_2^T = [l_1 \quad l_2 \quad \dots \quad l_{N_{max}}]$, and $G_{\rho_1, \rho_2} = \mathcal{T}_{2:(N_{max}+1), 2:(N_{max}+1)}^{(A2)}$ is the lower right block of $\mathcal{T}^{(A2)}$.

Proof: Appendix 5.7.6. ■

Remark 5.4.1. Corollary 5.4.1 provides an analytical expression for the stability boundary of A2 which has not been obtained in the earlier works [19] or [11]. It is reassuring that (5.4.22) agrees with numerical results of [19].

Remark 5.4.2. Using the proposed MJS approach, the condition that both ρ_1 and ρ_2 are strictly less than 1 is not necessary (which was needed in [19]).

5.4.3 Recover stability of A1

As aforementioned in Remark 5.3.1, A2 reduces to A1 when $\eta = 1$ and $\kappa_2 \equiv \kappa_1$. The probability transition of buffer content of A2 when $\eta = 1$ (i.e. A1 this case) is showed in Appendix A as matrix $\Pi^{(A1)}$. From Corollary 5.4.1, the stability for A1 is established as following:

Corollary 5.4.2 (Stochastic stability of A1).

The E-SAC scheme A1 yields a stochastically stable loop, in the sense that $V(x_k)$ satisfies the bound condition in Theorem 5.3.1, if $\mathcal{T}^{(A1)} \triangleq \text{diag}\{\alpha, \underbrace{\rho_1, \dots, \rho_1}_{N_{max}-1 \text{ times}}\} \Pi^{(A1)}$ is Schur stable.

Futher, if $\rho_1 < 1$ the Schur stability condition for $\mathcal{T}^{(A1)}$ is

$$\Omega \triangleq l_0 \alpha (1 + \rho_1 \Theta^T (I - \rho_1 G)^{-1} E_1) < 1 \quad (5.4.23)$$

where $\Pi^{(A1)}$ is showed in eq. (5.3.9), $\Theta^T = [l_1 \ l_2 \ \dots \ l_{N_{max}}]$, $E_1 = [1 \ 0 \ 0 \ \dots \ 0]^T \in \mathbb{R}^{N_{max} \times 1}$, $G = \Pi_{[2:(N_{max}+1); 2:(N_{max}+1)]}^{(A1)} \subset \mathbb{R}^{N_{max} \times N_{max}}$ is the lower right block of $\Pi^{(A1)}$ obtained by eliminating the first row and the first column.

Remark 5.4.3. Interestingly, (5.4.23) is the same stability condition as already derived with the State-dependent Random-time Drift condition method in [11].

Remark 5.4.4. The stability of A1 is independent of the triggering threshold d as showed in [11]. Similarly, the stability of A2 showed in Corollary 5.4.1 is also independent of d . The threshold d does however determine the size of the region that the system state converges to. In detail, in Theorem 1, the size of this region is $C_2 = \frac{1}{(\zeta_{\min})(1-\xi)} (\max\{\zeta_{\min} D, |\zeta_{\max} \sigma_{(-1)} - \xi \zeta_{\min}| D\})$. For E-SAC schemes A1 and A2, $D = \varphi_2(d)$ (as defined in Section 5.4.1) influences C_2 .

5.5 Numerical simulation

Given a plant with dynamics

$$x_{k+1} = -x_k + 0.1 \sin(x_k) + u_k + w_k \quad x_0 = 20 \quad (5.5.24)$$

where the disturbance w_k is i.i.d., normally distributed with zero mean and unit variance. For the proposed schemes with two control laws, the following control laws are adopted

$$\begin{cases} \kappa_1(x_k) = x_k - 0.1 \sin(x_k) + c_1 |x_k|, \\ \kappa_2(x_k) = x_k - 0.1 \sin(x_k) + c_2 |x_k|. \end{cases}$$

where $c_1 = 0.9$, $c_2 = 0.7$.

By choosing $V(x) = |x|$, it follows that $\rho_1 = 0.9$ in (5.1.6) and $\rho_2 = 0.7$ in (5.1.7).

It is assumed that evaluating κ_2 takes twice as many processing units than κ_1 (i.e., $\eta = 2$ in Assumption 5.1.2). Assume that the buffer size $\Lambda = 4$ and that the maximum available

processing units are $N_{max} = 4$.

The probability of successful transmission is given by:

$$q = \mathbf{Pr}\{\gamma_k = 1 \mid |x_k| > d\} = 0.5$$

For $N_k \in \{0, 1, 2, \dots, 4\}$, assume that the probabilities $p_j = \mathbf{Pr}\{N_k = j \mid \gamma_k = 1\}$ are equal for each $j \in \{0, 1, 2, \dots, 4\}$, i.e., $p_0 = p_1 = \dots = p_4 = 0.2$.

5.5.1 Performance

Performance is evaluated through the closed-loop empirical cost

$$J \triangleq \frac{1}{500} \left(\sum_{k=201}^{700} x_k^2 \right)$$

and the Channel Utilisation(%)

$$CU = \frac{\text{Total number of time steps at which } \gamma_k \neq 2}{\text{Total number of time steps}}$$

by averaging over 10^4 realizations.

Performance of different schemes are shown in Fig. 5.8. For a given threshold d (which is a design parameter) it is interesting that the no-buffering two-control law scheme $B2$ outperforms, in term of smaller empirical closed-loop cost and lower channel utilization, the one-control law scheme with buffer $A1$. The two-control law scheme with buffer $A2$ further outperforms $B2$ as expected and is the best scheme.

Obviously, the performance of two-control law schemes $B2$ and $A2$ depends on the parameters ρ_2 and η . Such dependence is currently a research topic.

5.5.2 Stability regions

For ease of comparison, ϵ that represents the ratio between the shrinking factors of control laws κ_1 and κ_2 is defined as:

$$\epsilon = \frac{\rho_2}{\rho_1}$$

Note that $\epsilon \in [0, 1]$, and it can be said that the smaller the ϵ is, the “finer” the second control law κ_2 is.

Figure 5.8: Performance, in term of a trade-off curve between channel utilisation CU and empirical cost J .

Figure 5.9: Boundaries of stability: algorithms $B1$ (red, circle) and $B2$ (blue, with $\epsilon = 1$ (dashed), $\epsilon = 0.8$ (dotted), $\epsilon = 0.5$ (dash-dot), $\epsilon = 0.05$ (solid)).

Control schemes without buffer

Fig. 5.9 shows the stability boundaries ($\Gamma = 1$ in Theorem 4.1 in [19] and Theorem 1 in [11]) of algorithms $B1$ and $B2$.

Fig. 5.9 shows that the no-buffering two-control law scheme $B2$ has a larger region with guaranteed stability as per our current results (the left area where $\Gamma < 1$) than the no-buffering one-control law $B1$ algorithm. Note that when $\epsilon = 1$ the boundaries corresponding to algorithms $B1$ and $B2$ coincide.

Control schemes with buffer

Fig. 5.10 shows the stability boundaries of $A2$ ($\Psi = 1$ in (5.4.22)) for different values of the parameter $\epsilon = 1, 0.8, 0.5, 0.05$, and the stability boundary of the one-control law with buffering $A1$ ($\Omega = 1$ in (5.4.23)). Fig. 5.10 illustrates that the stability boundary of the $A2$ algorithm depends on how good the fine control law κ_2 is. From this figure, it can be seen that $A2$ can handle a more unstable plant, provided κ_2 is good enough.

5.5.3 Stability region with respect to the computational cost of fine policy

The performance of $A2$ depends on: (1) how much does the fine policy cost, in terms of computational cost η processing units, and (2) how does the fine policy perform, in terms of smaller ρ_2 ? This section shows the region for stability guarantee of $A2$ with different values of η .

Consider another system with a greater open-loop growth than (5.5.24)

$$x_{k+1} = -1.34x_k + 0.01\sin(x_k) + u_k + w_k, \quad x_0 = 20 \quad (5.5.25)$$

and the two control laws

$$\kappa_1(x_k) = 1.34x_k - 0.01\sin(x_k) + 0.9|x_k|, \quad (5.5.26)$$

$$\kappa_2(x_k) = 1.34x_k - 0.01\sin(x_k) + c_2|x_k|. \quad (5.5.27)$$

where c_2 is decided later.

By choosing $V(x) = |x|$, it follows that $\rho_1 = 0.9$ in (5.1.6) and $\rho_2 = c_2$ in (5.1.7).

Fig. 5.11 shows the region for the open-loop bound (α) and closed-loop contractions (ρ_1 & $\rho_2 = \epsilon\rho_1$) that $A2$ and $A1$ guarantee to yield a stochastically stable system, i.e. the region is represented by eq. (5.4.22) and (5.4.23) given that $\rho_1 \in [0, 1]$. For a specific value of ρ_1 and ϵ , one can figure out the maximum open-loop bound α that is allowed so that the system is guaranteed to be stochastically stable. The region when using $A2$ depends on parameters η and ϵ . Point $Q1 = (1.3525, 0.9)$ is on the curve with label “ $A2 : \eta = 2, \epsilon = 0.5$ ” (dashed line with triangle). This implies that given two control laws:

Figure 5.10: Boundaries of stability: $A1$ (red, square) and $A2$ (blue with $\epsilon = 1$ (dashed), $\epsilon = 0.8$ (dotted), $\epsilon = 0.5$ (dash-dot), $\epsilon = 0.05$ (solid)).

(1) the coarse control law with $\rho_1 = 0.9$ and (2) the fine control law with parameters $\eta = 2$ and $c_2 = \rho_2 = 0.5 * 0.9 = 0.45$, the algorithm $A2$ yields a stochastically stable system, **if** the open-loop bound α satisfies $\alpha < 1.3527$. Since system (5.5.25) has $\alpha = 1.35 < 1.3527$, $A2$ yields a stochastically stable system with this configuration of two control laws. The blue line in Fig. 5.12 confirms this, as the averaged value of $V(x_k)$ is bounded.

Similarly, by looking at point $Q2$, it shows that $A2$ with $\rho_1 = 0.9$, $\rho_2 = 0.5 * 0.9 = 0.45$, $\eta = 3$ only guarantees to yield a stochastically stable system if the open-loop bound α satisfies $\alpha < 1.266$. And by looking at point $Q3$ on the curve labelled as $A1$, it shows that the one control law $A1$ with $\rho_1 = 0.9$ only guarantees to be able to stochastically stabilise a system if the open-loop bound α satisfies $\alpha < 1.175$. Indeed, the averaged value of $V(x_k)$ can be very large in these two cases, see Fig. 5.12 the triangle points and dotted line, since the open-loop bound of (5.5.25) is $\alpha = 1.35$ bigger than the allowed open-loop bounds of these two configurations.

Remark 5.5.1. Eq. (5.4.22) and (5.4.23) are sufficient conditions for stochastic stability of $A2$ and $A1$, respectively. Currently, necessary conditions are not available for these schemes.

Figure 5.11: Stability guarantee region for $A2$ with respect to η (processing units needed to compute κ_2) and $\epsilon = \frac{\rho_2}{\rho_1}$. Two-control scheme $A2$ with configuration $Q1$ guarantees to yield a stable system if the open-loop bound satisfying $\alpha < 1.3527$. Two-control scheme $A2$ with configuration $Q2$ guarantees to yield a stable system if $\alpha < 1.266$. One-control scheme $A1$ with configuration $Q3$ guarantees to yield a stable system if $\alpha < 1.175$. System (5.5.25) has open-loop bound $\alpha = 1.35$.

5.6 Conclusion

In this thesis, event-triggered anytime control algorithm is extended to schemes which feature multiple control laws. The case of the two control laws is presented in detail. While buffering is a good technique to better utilize the excess computational resource which may become available in an a priori unknown manner, introducing a fine control law give additional performance benefits. In fact, by using a coarse and a fine control law (without buffering), significant performance gains can be achieved. Adding buffer in the proposed two-control law scheme could further improve performance, if there are still slacks in the system that can be exploited. A general model and a novel stability analysis method for event-triggered sequence-based anytime schemes based on Markov jump systems ideas are developed. The proposed method is, unlike the State-dependent Random-time Drift condition approach, scalable for sophisticated schemes. Numerical simulation also shows that the stability region can be enlarged with the proposed two-control law schemes.

Figure 5.12: Averaged value of $V(x) = |x_k|$ over 10^4 random realizations.

5.7 Appendix

Appendix 5.7.1. Probability transition in A1 and A2

For ease of notation, we denote $\pi_{ij} = \pi_{ij}^{(A1)}$. We have

$$\begin{aligned} \pi_{ij} &= \mathbf{Pr}\{\theta_k = S_j | \theta_{k-1} = S_i, \gamma_k \neq 2, \gamma_{k-1} \neq 2\} = \mathbf{Pr}\{\theta_k = j-1 | \theta_{k-1} = i-1, \gamma_k \neq 2, \gamma_{k-1} \neq 2\} \\ &= \mathbf{Pr}\{N_k = j-1 | \gamma_k \neq 2\} = l_{j-1}, \\ &\forall \{(i, j) | 1 \leq i \leq N_{max} + 1, i \leq j \leq N_{max} + 1\} \cup \{(i, j) | 3 \leq i \leq N_{max}, 1 \leq j \leq i-2\} \end{aligned}$$

$$\begin{aligned} \pi_{21} &= \mathbf{Pr}\{\theta_k = -1 | \theta_{k-1} = 1, \gamma_k \neq 2, \gamma_{k-1} \neq 2\} = \mathbf{Pr}\{N_k = 0 | \gamma_k \neq 2\} = l_0 \\ \pi_{i(i-1)} &= \mathbf{Pr}\{\theta_k = i-2 | \theta_{k-1} = i-1, \gamma_k \neq 2, \gamma_{k-1} \neq 2\} \\ &= \mathbf{Pr}\{N_k = 0 | \gamma_k \neq 2\} + \mathbf{Pr}\{N_k = i-2 | \gamma_k \neq 2\} = l_0 + l_{i-2}, \forall 3 \leq i \leq N_{max} \end{aligned}$$

where l_j is defined as (5.3.10).

Similarly to the algorithm A1 analysed above, one can show that:

$$\begin{aligned} \pi_{i1} &= l_0, \quad i \in \{1, 2, \eta + 1\}, \quad \pi_{i1} = 0 \quad \forall i \neq \{1, 2, \eta + 1\} \\ \pi_{ij} &= l_{j-1} + l_0, \quad \forall j = i - \eta, \quad i = \eta + 2, \dots, N_{max} \\ \pi_{ij} &= l_j \quad \text{otherwise} \end{aligned}$$

Further, the probability transition matrix is as follows:

$$\Pi^{(A2)} = \begin{bmatrix} l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 + l_0 & l_2 & \cdots & \cdots & l_{N_{max}} \\ \cdots & \cdots & \ddots & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 & \cdots & l_{N_{max}-2} + l_0 & \cdots & l_{N_{max}} \end{bmatrix},$$

if $\eta = 2$.

For $\eta > 2$, we have:

$$\Pi^{(A2)} = \begin{bmatrix} l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 + l_0 & l_2 & \cdots & \cdots & l_{N_{max}} \\ \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ 0 & l_1 & \cdots & l_{\eta-2} + l_0 & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 + l_0 & l_2 & \cdots & \cdots & l_{N_{max}} \\ \cdots & \cdots & \ddots & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 & \cdots & l_{N_{max}-\eta} + l_0 & \cdots & l_{N_{max}} \end{bmatrix}$$

Appendix 5.7.2. Proof of Lemma 5.3.1

If $N_k > 0$, then u_k is determined by the current state x_k . If the processor is not available, then u_k has been determined by the states which are at most Λ time stages old, or is given by $u_k = 0$. Since the processor availability is independent of the state, the stochastic process $\{y\}_{\mathbb{N}_0}$ is Markovian.

Appendix 5.7.3. Proof of Theorem 5.3.1

Proof. We define $\zeta_{\max} = \max\{\zeta_{(i)}\}_{i=1,\dots,M}$, $\zeta_{\min} = \min\{\zeta_{(i)}\}_{i=1,\dots,M}$, i.e., we have

$$0 < \zeta_{\min} \leq \zeta_{(\theta_k)} \leq \zeta_{\max}, \quad \forall \theta_k = 1, \dots, M. \quad (5.7.28)$$

Consider

$$J_k \triangleq J(y_k, \theta_k) = \zeta_{(\theta_k)} \tilde{V}(y_k) = \zeta_{(\theta_k)} \tilde{V}_k$$

where θ_k is as in Eq. (5.3.17).

We will next establish the drift condition

$$\mathbf{E}\{J_{k+1}|y_k, \theta_k\} \leq \bar{D} + \xi J(y_k, \theta_k)$$

Since $J(y_k, \theta_k)$ becomes undefined when $\theta_k = -1$ (deterministic mode, $\zeta_{(\theta_k=-1)}$ does not exist), without loss of generality, we take (extended value) $\zeta_{(-1)} = \zeta_{\min}$.

If $\theta_k = -1$, then $\beta_k = 0$, i.e., the deterministic mode is active. We have $\tilde{V}(y_k) \leq D$ (Assumption 5.3.2), and we also have $\zeta_{(\theta_{k+1})}\tilde{V}_{k+1} \leq \zeta_{\max}\sigma_{(-1)}\tilde{V}_k$. From $\tilde{V}(y_k) \leq D$, we obtain $\zeta_{\max}\sigma_{(-1)}\tilde{V}_k \leq \xi\zeta_{\min}\tilde{V}_k + |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D \leq \xi J(y_k, \theta_k) + |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D$. Then it follows that

$$\begin{aligned} \mathbf{E}\{J(y_{k+1}, \theta_{k+1})|y_k, \theta_k = 0\} &= \mathbf{E}\{\zeta_{(\theta_{k+1})}\tilde{V}_{k+1}|y_k, \theta_k = 0\} \\ &\leq \xi J(y_k, \theta_k) + |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D \end{aligned} \quad (5.7.29)$$

If $\theta_k = i > 0$, then $\beta_k = 1$. By denoting $\Upsilon_k \triangleq (y_k, \theta_k = i > 0)$, using the law of total expectation we have:

$$\begin{aligned} \mathbf{E}\{J(y_{k+1}, \theta_{k+1})|y_k, \theta_k = i > 0\} &= \\ \mathbf{E}\{J(y_{k+1}, \theta_{k+1})|\Upsilon_k, \beta_{k+1} = 0\}\mathbf{Pr}\{\beta_{k+1} = 0|\Upsilon_k\} + & \end{aligned} \quad (5.7.30)$$

$$\begin{aligned} &\mathbf{E}\{J(y_{k+1}, \theta_{k+1})|\Upsilon_k, \beta_{k+1} = 1\}\mathbf{Pr}\{\beta_{k+1} = 1|\Upsilon_k\} \\ &\leq \mathbf{E}\{J(y_{k+1}, \theta_{k+1})|\Upsilon_k, \beta_{k+1} = 0\} + \mathbf{E}\{J(y_{k+1}, \theta_{k+1})|\Upsilon_k, \beta_{k+1} = 1\} \end{aligned} \quad (5.7.31)$$

For $\beta_{k+1} = 0$, this implies that $\theta_{k+1} = -1$. And by Assumption 5.3.2, we have $\tilde{V}_{k+1} \leq D$, then

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1})|\Upsilon_k, \beta_{k+1} = 0\} = \mathbf{E}\{\zeta_{(-1)}\tilde{V}_{k+1}|\Upsilon_k, \beta_{k+1} = 0\} \leq \zeta_{\min}D \quad (\zeta_{(-1)} = \zeta_{\min}) \quad (5.7.32)$$

For the case $\beta_{k+1} = 1$, the stochastic controller is deployed, i.e., $\theta_{k+1} > 0$, we also have $\beta_k = 1$, since $\theta_k > 0$.

We have, using the law of total expectation and using (5.3.18),

$$\begin{aligned}
& \mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 1\} \\
&= \sum_{j=1}^M \mathbf{Pr}\{\theta_{k+1} = j | \theta_k = i, y_k, \beta_k = 1, \beta_{k+1} = 1\} \\
&\quad \mathbf{E}\{\tilde{V}_{k+1} \zeta_{(\theta_{k+1})} | \theta_{k+1} = j, y_k, \theta_k = i, \beta_k = 1, \beta_{k+1} = 1\} \\
&\leq \sum_{j=1}^M (\pi_{ij}) (\sigma_{(i)} \tilde{V}_k \zeta_{(j)}) = \tilde{V}_k (\zeta_{(i)} - \nu_{(i)}) \quad (\text{using (5.3.18)}) \\
&= \tilde{V}_k \zeta_{(i)} \left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) = J(y_k, \theta_k) \left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) \\
&\leq \xi J(y_k, \theta_k)
\end{aligned} \tag{5.7.33}$$

since

$$\left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) \leq 1 - \frac{\min_{1 \leq j \leq M} \{\nu_{(j)}\}}{\max_{1 \leq k \leq M} \{\zeta_{(k)}\}} \triangleq \xi, \quad \forall i = 1, \dots, M$$

and $0 < \xi < 1$ due to $\zeta_{(k)} > \nu_{(k)} > 0$, $\forall k = 1, 2, \dots, M$ (from (5.3.18)).

Expressions (5.7.31)-(5.7.33) lead to:

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | y_k, \theta_k > 0\} \leq \zeta_{\min} D + \xi J(y_k, \theta_k) \tag{5.7.34}$$

From (5.7.29) and (5.7.34), we obtain

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | y_k, \theta_k\} \leq \bar{D} + \xi J(y_k, \theta_k)$$

i.e., we have

$$\mathbf{E}\{J(z_{k+1}) | z_k\} \leq \bar{D} + \xi J(z_k) \tag{5.7.35}$$

where

$$\begin{aligned}
z_k &\triangleq (y_k, \theta_k) \\
\bar{D} &= \max\{\zeta_{\min} D, |\zeta_{\max} \sigma_{(-1)} - \xi \zeta_{\min}| D\}
\end{aligned}$$

From (5.7.35) and using the Markovian property of $\{z_k\}$,ⁱ we have

$$\mathbf{E}\{J(z_1) | z_0\} \leq \bar{D} + \xi J(z_0) \tag{5.7.36}$$

$$\mathbf{E}\{J(z_2) | z_1\} = \mathbf{E}\{J(z_2) | z_1, z_0\} \leq \bar{D} + \xi J(z_1) \tag{5.7.37}$$

ⁱSince $\{y_k\}$ and $\{\theta_k\}$ are Markovian, see Lemma 5.3.1 and (5.3.17).

Taking expectation $\mathbf{E}\{.\mid z_0\}$ of both sides of (5.7.37), using tower property of expectation and using (5.7.36) we have

$$\begin{aligned}\mathbf{E}\{J(z_2)\mid z_0\} &= \mathbf{E}\{\mathbf{E}\{J(z_2)\mid z_1, z_0\}\mid z_0\} \leq \bar{D} + \xi \mathbf{E}\{J(z_1)\mid z_0\} \\ &\leq \bar{D} + \xi \bar{D} + \xi^2 J(z_0)\end{aligned}$$

By iterating the above procedure, we obtain

$$\mathbf{E}\{J(z_k)\mid z_0\} \leq \bar{D} \left(\sum_{i=0}^{k-1} \xi^i \right) + \xi^k J(z_0) \leq \frac{\bar{D}}{1-\xi} + \xi^k J(z_0)$$

i.e.,

$$\mathbf{E}\{J(y_k, \theta_k)\mid y_0, \theta_0\} \leq \frac{\bar{D}}{1-\xi} + \xi^k J(y_0, \theta_0)$$

Using the law of total expectationⁱⁱ, if y_0 is a discrete random variable we have

$$\begin{aligned}\mathbf{E}\{J(y_k, \theta_k)\} &= \sum_{\forall(y_0, \theta_0)} \mathbf{E}\{J(y_k, \theta_k)\mid y_0, \theta_0\} \mathbf{Pr}\{y_0, \theta_0\} \\ &\leq \sum_{\forall(y_0, \theta_0)} \left(\frac{1}{1-\xi} \bar{D} + \xi^k J(y_0, \theta_0) \right) \mathbf{Pr}\{y_0, \theta_0\} \\ &= \frac{1}{1-\xi} \bar{D} + \sum_{\forall(y_0, \theta_0)} \left(\xi^k J(y_0, \theta_0) \right) \mathbf{Pr}\{y_0, \theta_0\} = \frac{1}{1-\xi} \bar{D} + \xi^k \mathbf{E}\{J(y_0, \theta_0)\}\end{aligned}\quad (5.7.38)$$

If y_0 is a continuous random variable, we haveⁱⁱⁱ

$$\begin{aligned}\mathbf{E}\{J(y_k, \theta_k)\} &= \int_{y_0} \mathbf{E}\{J(y_k, \theta_k)\mid y_0, \theta_0\} \mathbf{pdf}\{y_0, \theta_0\} dy_0 \\ &\leq \int_{y_0} \left(\frac{1}{1-\xi} \bar{D} + \xi^k J(y_0, \theta_0) \right) \mathbf{pdf}\{y_0, \theta_0\} dy_0 \\ &= \frac{1}{1-\xi} \bar{D} + \int_{y_0} \left(\xi^k J(y_0, \theta_0) \right) \mathbf{pdf}\{y_0, \theta_0\} dy_0 = \frac{1}{1-\xi} \bar{D} + \xi^k \mathbf{E}\{J(y_0, \theta_0)\}\end{aligned}\quad (5.7.39)$$

Since $\zeta_{\max} \tilde{V}_k \geq J(y_k, \theta_k) = \zeta_{(\theta_k)} \tilde{V}_k \geq \zeta_{\min} \tilde{V}_k$, then $\mathbf{E}\{J(y_k, \theta_k)\} \geq \mathbf{E}\{\tilde{V}_k\} \zeta_{\min}$ and $\mathbf{E}\{J(y_0, \theta_0)\} \leq \zeta_{\max} \mathbf{E}\{\tilde{V}_0\}$. Then, from (5.7.38) (or (5.7.39)) we have

$$\zeta_{\min} \mathbf{E}\{\tilde{V}_k\} \leq \zeta_{\max} \xi^k \mathbf{E}\{\tilde{V}_0\} + \frac{1}{1-\xi} \bar{D}$$

□

ⁱⁱHere we assumed θ_0 is known.

ⁱⁱⁱ**pdf**: probability density function.

Appendix 5.7.4. Proof of Corollary 5.3.1

Since $\mathcal{T}$ is Schur stable, it follows that

$$(I - \mathcal{T})^{-1} = I + \sum_{i=1}^{\infty} (\mathcal{T})^i = I + \bar{\mathcal{T}}$$

where $\bar{\mathcal{T}} = \sum_{i=1}^{\infty} (\mathcal{T})^i$.

As all entries of $\mathcal{T}$ are non-negative, all entries of $\bar{\mathcal{T}}$ are non-negative.

Thus, for any given $\underline{\nu} \succ 0$, the solution of (5.3.19) is $\underline{\zeta} = (I - \mathcal{T})^{-1} \underline{\nu} = \underline{\nu} + \bar{\mathcal{T}} \underline{\nu}$ satisfying $\underline{\zeta} \succ 0$. Then, by applying Theorem 1, we obtain the desired bound for $\mathbf{E}\{\tilde{V}_k\}$. ■

Appendix 5.7.5. Prove of Lemma 5.4.1

Consider the characteristic polynomial $P(\lambda) \triangleq \det(\lambda I - H) = \begin{vmatrix} \lambda - X & -Y \\ -Z & \lambda I_m - M \end{vmatrix}$. We denote $\text{eig}(\mathcal{M})$ as an eigenvalue of a matrix $\mathcal{M}$.

For λ not inside the unit circle $|\lambda| \geq 1$, we have $\text{eig}(\lambda I_m - M) = \lambda - \text{eig}(M) \neq 0$ (due to Schur stability of M) hence $\lambda I_m - M$ is invertible. Using the determinant result for a 2-by-2 block matrix, we have

$$P(\lambda) = \det(\lambda I_m - M) \det(\lambda - X - Y(\lambda I_m - M)^{-1}Z) = h(\lambda)g(\lambda), \quad \forall |\lambda| \geq 1 \quad (5.7.40)$$

where $h(\lambda) = \det(\lambda I_m - M)$ and $g(\lambda) = \det(\lambda - X - Y(\lambda I_m - M)^{-1}Z)$.

“ $\Rightarrow$ ” We have H is Schur stable, now we need to proof $g(1) > 0$.

Since H is Schur stable, from the Schur-Cohn criteria (see [84], p.27) we have $P(1) > 0$. It follows that $h(1)g(1) > 0$. As M is Schur stable, we have $h(1) = \det(I_m - M) > 0$. Hence, $g(1) > 0$.

“ $\Leftarrow$ ” We have $g(1) > 0$, now we need to show that H is Schur stable. Assume that H is not Schur stable. Then there exists λ_0 , $|\lambda_0| \geq 1$ such that $P(\lambda_0) = 0$. We then have $\lambda_0 I_m - M$ is invertible and then $P(\lambda_0) = h(\lambda_0)g(\lambda_0) = 0$ where $h(\lambda_0) = \det(\lambda_0 I_m - M) \neq 0$. It follows that $g(\lambda_0) = 0$.

(1) If λ_0 is a real number: we can proof that $g(\lambda)$ is a strictly increasing function on $\mathbb{R}$. Then we have $0 = g(\lambda_0) > g(1) > 0$. This is a contradiction.

(2) If λ_0 is a complex number, we have λ_0^* , the conjugate of λ_0 , is also an eigenvalue of H . Then λ_0^2 and $(\lambda_0^*)^2$ are eigenvalues of H^2 . Then, $\text{trace}(H^2) = \sum(\text{eig}(H^2)) \geq \lambda_0^2 + (\lambda_0^*)^2 \geq 2$. It can also be shown that $\text{trace}(H^2) = X^2 + 2YZ + \text{trace}(M^2) < (X + YZ)^2 + \text{trace}(M^2) < 2$, since $g(1) > 0 \Rightarrow X + YZ < 1$ and $\text{trace}(M^2) < 1$ as in the assumption of Lemma 2. This is also a contradiction.

The two above contradictions establish the result. ■

Appendix 5.7.6. The certification matrix of A_2 , $\mathcal{T}^{(A_2)}$, is calculated as follows. From (5.4.21) and (5.3.20), it follows that $\Phi^{(A_2)} = \text{diag}\{\alpha, \underbrace{\rho_1, \dots, \rho_1}_{\eta-1 \text{ times}}, \underbrace{\rho_2, \dots, \rho_2}_{N_{max}-\eta+1 \text{ times}}\}$. Then, $\mathcal{T}^{(A_2)} \triangleq \Phi^{(A_2)} \Pi^{(A_2)}$. Then, the first statement of Corollary 5.4.1 is directly proved by Corollary 5.3.1.

To prove the second statement of Corollary 5.4.1, firstly we have $\mathcal{T}^{(A_2)}$ can be written as:

$$\mathcal{T}^{(A_2)} = \begin{bmatrix} l_0 \alpha & \alpha \Theta^T \\ l_0 \rho_1 E_1 & \rho_1 G \end{bmatrix}.$$

It is easy to prove that the (2,2) block of $\mathcal{T}^{(A_2)}$, which is G_{ρ_1, ρ_2} , is Schur stable by proving that $\|G_{\rho_1, \rho_2}\|_\infty < 1$, hence $1 > \|G_{\rho_1, \rho_2}\|_\infty \geq \sigma(G_{\rho_1, \rho_2})$ (spectral radius is less than or equal any matrix norm). It can also be shown that $\text{trace}(G_{\rho_1, \rho_2}^2) < 1$. Therefore, by applying Lemma 5.4.1 we have the Schur condition for $\mathcal{T}^{(A_2)}$ is that

$$1 - l_0 \alpha - l_0 \alpha \Theta_2^T (I - G_{\rho_1, \rho_2})^{-1} E_2 > 0$$

which gives (5.4.22).

Appendix 5.7.7. State-dependent Random-time Drift condition for A_2

Theorem 5.7.1 ([19]). Define

$$\Psi \triangleq \alpha \sum_{j \in \mathbb{N}} \sum_{m=0}^j \rho_1^m \rho_2^{j-m-1} \mathbf{Pr}\{\Delta_i = j, r_i = m | \gamma_{k_{i+1}}\} \neq 2\} \quad (5.7.41)$$

where r_i is the number of the times κ_1 is active between k_i and k_{i+1} , where $k_0 = 0$ and

$$k_{i+1} = \inf\{k \in \mathbb{N} : k > k_i, \theta_k = (0, 0)\}, i \in \mathbb{N}_0$$

($\theta_k = (0, 0)$: empty buffer) and denote $\mathbb{K} = \{k_i\}_{i \in \mathbb{N}_0}$. If Algorithm A_2 is used and $\Psi < 1$, then, for all $i \in \mathbb{N}$

$$\max_{k \in k_i, k_i+1, \dots, k_{i+1}-1} \mathbf{E}\{\varphi_1(|x_k|)\} \leq (1 + \alpha \frac{1}{1 - \rho_1}) \Psi^i \mathbf{E}\{\varphi_2(x_0)\} + \frac{\varphi_2(d)}{1 - \Psi} < \infty \quad (5.7.42)$$

Proof. Define $\Delta_i = k_{i+1} - k_i$ representing the time duration that the buffer become emptied again. For all $k_i \in \mathbb{K}$ and $l \in \{1, 2, \dots, \Delta_i - 1\}$:

$$\begin{cases} u_{k_i} = 0 \\ u_{k_i+l} \in \{\kappa_1(x_{k_i+l}), \kappa_2(x_{k_i+l})\}. \end{cases}$$

Therefore, by using $V(x, \kappa_i(x)) \leq \rho_i V(x)$ as in Assumption 5.1.2, it follows that

$$\mathbf{E}\{V(x_{k_{i+1}}) | x_{k_i} = \chi, \Delta_i = j, r_i = m\} \leq \alpha \rho_1^m \rho_2^{j-m-1} V(\chi) \quad , \quad \forall \chi \in \mathbb{R}^n$$

When the buffer is emptied by the trigger $\gamma_k = 2$ (within threshold event):

$$\mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} = 2\} < \varphi_2(d) \triangleq D$$

Using the law of total expectation, the definition of conditional probability and the assumptions made:

$$\begin{aligned} \mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi\} &= \mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} = 2\} \times \mathbf{Pr}\{\gamma_{k_{i+1}} = 2|x_{k_i} = \chi\} \\ &\quad + \mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \times \mathbf{Pr}\{\gamma_{k_{i+1}} \neq 2|x_{k_i} = \chi\} \\ &\leq \varphi_2(d) + E\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \times \mathbf{Pr}\{\beta(k_i + 1) \neq 2|x_{k_i} = \chi\} \\ &= D + \sum_{j \in \mathbb{N}} \mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2, \Delta_i = j\} \times \mathbf{Pr}\{\Delta_i = j|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \\ &= D + \sum_{j \in \mathbb{N}} \sum_{m=0}^j \mathbf{E}\{V(x_{k_{i+1}})|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2, \Delta_i = j, r_i = m\} \\ &\quad \times \mathbf{Pr}\{\Delta_i = j|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \times \mathbf{Pr}\{r_i = m|\Delta_i = j, x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \\ &\leq D + \sum_{j \in \mathbb{N}} \sum_{m=0}^j \alpha \rho_1^m \rho_2^{j-m-1} V(\chi) \times \mathbf{Pr}\{\Delta_i = j, r_i = m|x_{k_i} = \chi, \gamma_{k_{i+1}} \neq 2\} \\ &= \varphi_2(d) + \Psi V(\chi), \forall \chi \in \mathbb{R}^n \end{aligned}$$

with Ψ defined as in (5.7.41). Note that, the definition of $\mathbb{K} = \{k_i\}_{i \in \mathbb{N}_0}$ gives that $\{x\}_{\mathbb{K}}$ is Markovian [11], giving

$$\mathbf{E}\{V(x_{k_i})|x_{k_0} = \chi\} \leq \Psi^i V(\chi) + \frac{\varphi_2(d)}{1 - \Psi}$$

Recall that $0 < \rho_2 < \rho_1 < 1$, and with Assumptions 1 and 4, it follows that

$$V(x_{k_{i+1}}) \leq \alpha V(x_{k_i}) \tag{5.7.43}$$

$$V(x_{k+1}) \leq \rho_1 V(x_k), \quad \forall k \notin \mathbb{K} \tag{5.7.44}$$

Then, it follows that

$$\begin{aligned} &\mathbf{E} \left\{ \sum_{k=k_i}^{k_{i+1}-1} V(x_k) \mid x_{k_i} = \chi, \Delta_i = j \geq 2 \right\} \\ &\leq \left(1 + \alpha \sum_{l=0}^{j-2} \rho_1^l \right) V(\chi) + \frac{\varphi_2(d)}{1 - \Psi} \leq \left(1 + \alpha \frac{1}{1 - \rho_1} \right) V(\chi) + \frac{\varphi_2(d)}{1 - \Psi} \end{aligned}$$

Then, by using a similar method to that employed in [85] (Thm.5.3) it follow that:

$$\mathbf{E} \left\{ \sum_{k=k_i}^{k_{i+1}-1} V(x(k)) \mid x(k_0) = \chi \right\} \leq (1 + \alpha \frac{1}{1 - \rho_1}) \Psi^i V(\chi) + \frac{\varphi_2(d)}{1 - \Psi}, \forall i \in \mathbb{N}$$

By the law of total expectation and with $\Psi < 1$, (5.7.42) is obtained. $\square$

Remark 5.7.1. It is demonstrated in [19] that the calculation of $\mathbf{Pr}\{\Delta_i = j, r_i = m | \gamma_{k_{i+1}} \neq 2\}$ (number of time each control policy active during the a time interval) used in (5.7.41) is a combinatoric-nature problem. This problems quickly becomes intractable when the time interval tends to infinite as well as the number of control laws increases. As a result, an analytical expression for the stability condition (5.7.41) has not been obtained by the SRD approach.

Chapter 6

Conclusions and future works

6.1 Conclusion

This thesis addresses the challenges when extending Model Predictive Control to networked or embedded systems: (1) accelerating the solution of MPC optimisation problem and (2) dealing with the time-varying computational resources. A novel method for exploiting banded null basis is presented for accelerating the computation of MPC. Using the banded basis reduces the KKT system to two system of linear equations with (i) smaller dimension and (ii) banded matrices. Banded null bases can also be used to transform a sparse QP to a reduced-Hessian QP while maintaining banded matrices. When testing with ADMM and implementing the improved ADMM algorithm on an industrial device with a wide range of benchmark problems, results confirm that the banded null bases technique outperforms existing approaches with a significant reduction of runtime. To deal with the time-varying computational resources effectively, event-triggered sequence-based anytime (E-SAC) control with multiple control laws are proposed, and analysis framework for the new E-SAC schemes has been developed. More details are given as follows:

- Methods for exploiting a banded basis including: (1) to solve the KKT system with a banded basis and (2) to obtain the banded reduced-Hessian QP problem. The proposed method is tested with ADMM algorithms. Implementation results with a wide range of MPC problems on an industrial device (ARM9, 400MHz, 64MB RAM), show that significant reduction of the runtime is achieved, up to 26 times, compared with a standard implementation.
- Implementation on the ARM Cortex A9 processor (667MHz, 1GB RAM) for a 3-input 9-state MPC problem shows that ADMM-based algorithm accelerated by using the banded basis is 3 times faster than CVXGEN a well-performing interior-point

method. In addition, the code size is small, and the time-per-iteration grows linearly with respect to the prediction horizon. This suggests that implementing an ADMM-based MPC for non-trivial problems on moderate microcontrollers is a viable task.

- For a single-input system, an algorithm for efficiently computing the banded null basis is proposed; and the existence of a banded basis in MPC is proved with deriving an upper bound for its sparsity. For multi-input systems, a conjecture is raised; however, the existence of a banded basis in 18 MPC problems in the literature and random-generated systems has been confirmed by numerical tests.
- Methods for trading sparsity for conditioning of the banded basis are proposed to prevent numerical error due to the ill-conditioned issue that may arise in some cases.
- An ADMM algorithm that can handle the polytopic constraints in the sparse QP formulation is proposed. This algorithm becomes division-free when some matrix inversion is computed offline, enabling a fixed-point arithmetic on FPGA.
- Event-triggered anytime control is extended to schemes featuring multiple control laws. The proposed schemes yield a better performance in terms of empirical cost, channel utilization and region for stability guarantee compared with the basic E-SAC.
- A new stability analysis method based on Markov jump system is developed for the multiple control law scheme. The proposed method systematically extends to a more sophisticated E-SAC scheme for which, until now, no analytical expression had been obtained.

6.2 Future work

The banded null basis is the new technique that has not been employed in existing MPC-related algorithms. This technique can potentially be incorporated into existing optimisation algorithms, e.g. interior-point methods and splitting methods. Future work related to banded basis techniques could be

- Prove the existence of a banded null basis in multi-input systems. This may be accomplished by exploiting the controllability indices of the linear system.
- Accelerate the computation of interior-point methods and fast dual forward-backwards method based on using banded null basis. As mentioned in Chapter 2, these optimisation methods also require to solve the system of linear equations which are similar to the system of linear equations of ADMM. Therefore, the presented banded null basis framework can be extended directly.

- Investigate the applicability of banded basis techniques in active-set methods, operator splitting methods. The system of linear equations also arises in these methods, but in different formats and may be banded. In general, a banded matrix may have a banded basis for its null space. Hence, banded null basis method may also bring advantages to these methods.
- Extend the banded null basis technique to nonlinear MPC problems. In nonlinear MPC problem, the equality-constraint matrix may change each sampling instant as the result of the linearisation. Hence, the banded basis needs to be computed online. Then, it is worth to investigate methods to compute the banded basis efficiently in this scenario.
- To systematically solve the optimisation problem used in the method to trade sparsity for conditioning, e.g. to convert to a convex optimisation problem and then to employ existing tools that have been developed in the convex optimisation literature.
- Use QR decomposition to solve the system of linear equations when the banded null basis is ill-conditioned, as QR decomposition is more numerical robustness than Cholesky factorisation.
- Complete a banded null basis-based MPC toolbox and release to the public.

For the event-triggered anytime control topic, further work could be

- Currently, this thesis only addresses the stochastic stability. It is useful to derive a performance analysis framework for E-SAC to systematically compare the performance of different schemes.
- Current strategy of utilising the processor availability is that the E-SAC algorithms use up the given processor availability. If the E-SAC algorithms could free the processor early, power consumption is saved. Therefore, it is worth to investigate strategy to reduce processor utilisation while guaranteeing satisfactory performance. The direction could be to solve an optimisation problem to obtain the probability transition matrix describing the processor availability, then to force the processor to run in the obtained characteristics.
- The current stability analysis assumes that there is no process noise and model uncertainty which affect the quality of tentative control inputs. Hence, it is worth to extend the current framework taking into account them. One may make assumption about the bounded property of the uncertainties and then analyse the deviation of the state prediction and the deviation of the Lyapunov function.

- Analyse E-SAC with Markovian processor/sensor availability scenarios, as the proposed analysis framework can be systematically extended to these cases. To do this, one only needs to work out the probability transition matrix under these scenarios, then apply the presented framework to obtain the result.
- Investigate a suitable implementation of MPC that fits the principle and model of multi-control law E-SAC schemes, e.g. E-SAC schemes for MPC with a short horizon and long horizon strategy could be a candidate.

Bibliography

- [1] P. Kapasouris, M. Athans, and G. Stein, “Design of feedback control systems for stable plants with saturating actuators,” in *Proc. IFAC Symp. on Nonlinear Control System Design*. Pergamon Press, 1990, pp. 302–307.
- [2] A. Bemporad, A. Casavola, and E. Mosca, “Nonlinear control of constrained linear systems via predictive reference management,” *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 340–349, 1997.
- [3] W. Ding, M. R. Ganesh, R. N. Severinghaus, J. J. Corso, and D. Panagou, “Realtime model predictive control for keeping a quadrotor visible on the camera field-of-view of a ground robot,” in *Proc. American Control Conference (ACC)*, 2016, pp. 2259–2264.
- [4] J. Mattingley and S. Boyd, “CVXGEN: A code generator for embedded convex optimization,” *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.
- [5] J. Richalet, A. Rault, J. Testud, and J. Papon, “Model predictive heuristic control: Applications to industrial processes,” *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.
- [6] C. R. Cutler and B. L. Ramaker, “Dynamic matrix control - a computer control algorithm,” in *AICHE national meeting*, 1979.
- [7] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
- [8] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.
- [9] M. Benzi, G. H. Golub, and J. Liesen, “Numerical solution of saddle point problems,” *Acta Numerica*, vol. 14, no. 1, pp. 1–137, 2005.
- [10] D. E. Quevedo and V. Gupta, “Sequence-based anytime control,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 377–390, Feb 2013.

- [11] D. E. Quevedo, V. Gupta, W.-J. Ma, and S. Yüksel, “Stochastic stability of event-triggered anytime control,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3373–3379, Dec 2014.
- [12] P. Scokaert, D. Mayne, and J. Rawlings, “Suboptimal model predictive control (feasibility implies stability),” *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, Mar 1999.
- [13] G. Constantinides, A. Kinsman, and N. Nicolici, “Numerical data representations for FPGA-based scientific computing,” *IEEE Design & Test of Computers*, vol. 4, no. 28, pp. 8–17, 2011.
- [14] Y. Saad and H. A. Van Der Vorst, “Iterative solution of linear systems in the 20th century,” *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 1–33, 2000.
- [15] J. L. Jerez, G. A. Constantinides, and E. C. Kerrigan, “An fpga implementation of a sparse quadratic programming solver for constrained predictive control,” in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*. ACM, 2011, pp. 209–218.
- [16] A. Domahidi, A. U. Zraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, “Efficient interior point methods for multistage problems arising in receding horizon control,” in *Proc. 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 668–674.
- [17] L. Standardi, N. Poulsen, J. Jorgensen, and L. Sokoler, “Computational efficiency of economic MPC for power systems operation,” in *Innovative Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES*, Oct 2013, pp. 1–5.
- [18] S. Yüksel and S. P. Meyn, “Random-time, state-dependent stochastic drift for Markov chains and application to stochastic stabilization over erasure channels,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 47–59, 2013.
- [19] K. Q. Huang, T. V. Dang, K. V. Ling, and D. E. Quevedo, “Event-triggered anytime control with two controllers,” in *Proceeding of The 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan*, Dec 2015, pp. 4157–4162.
- [20] J. M. Maciejowski, *Predictive control: with constraints*. Prentice Hall, 2002.
- [21] S. J. Wright, “Applying new optimization algorithms to model predictive control,” in *AIChE Symposium Series*, vol. 93, no. 316. Citeseer, 1997, pp. 147–155.
- [22] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

- [23] S. J. Wright, *Primal-dual interior-point methods*. Siam, 1997.
- [24] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, New York, 1987.
- [25] H. J. Ferreau, H. G. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [26] M. S. Lau, S. Yue, K. Ling, and J. Maciejowski, “A comparison of interior point and active set methods for FPGA implementation of model predictive control,” in *Proc. European Control Conference*, 2009, pp. 156–160.
- [27] S. Yue, K. V. Ling, and J. M. Maciejowski, “A FPGA implementation of model predictive control,” in *Proc. American Control Conference (ACC)*. Minneapolis, MN, IEEE, 2006, pp. 1930–1935.
- [28] E. Hartley, J. Jerez, A. Suardi, J. Maciejowski, E. Kerrigan, and G. Constantinides, “Predictive control using an fpga with application to aircraft control,” *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 3, pp. 1006–1017, May 2014.
- [29] H. Peyrl, H. J. Ferreau, and D. Kouzoupis, “A hybrid hardware implementation for nonlinear model predictive control,” in *5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015*.
- [30] P. Zhang, J. Zambreno, and P. H. Jones, “An embedded scalable linear model predictive hardware-based controller using admm,” in *2017 IEEE 28th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, July 2017, pp. 176–183.
- [31] B. Khusainov, E. C. Kerrigan, A. Suardi, and G. A. Constantinides, “Nonlinear predictive control on a heterogeneous computing platform,” in *IFAC World Congress 2017*. IFAC.
- [32] S. Richter, C. N. Jones, and M. Morari, “Real-time input-constrained MPC using fast gradient methods,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 7387–7393.
- [33] —, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” *Automatic Control, IEEE Transactions on*, vol. 57, no. 6, pp. 1391–1403, 2012.
- [34] P. Patrinos and A. Bemporad, “An accelerated dual gradient-projection algorithm for linear model predictive control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 662–667.

- [35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [36] P. Giselsson and S. Boyd, “Metric selection in fast dual forward–backward splitting,” *Automatica*, vol. 62, pp. 1–10, 2015.
- [37] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, “Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems,” *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 644–658, March 2015.
- [38] A. U. Raghunathan and S. Di Cairano, “Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection,” in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 4324–4329.
- [39] B. O’Donoghue, G. Stathopoulos, and S. P. Boyd, “A splitting method for optimal control.” *IEEE Trans. Contr. Sys. Techn.*, vol. 21, no. 6, pp. 2432–2442, 2013.
- [40] M. Annergren, A. Hansson, and B. Wahlberg, “An ADMM algorithm for solving l-1 regularized MPC,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4486–4491.
- [41] M. J. Koegel and R. Findeisen, “Parallel solutions of model predictive control using the alternating direction method of multipliers,” in *Nonlinear Model Predictive Control*, vol. 4, no. 1, 2012, pp. 369–374.
- [42] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, “Embedded online optimization for model predictive control at megahertz rates,” *arXiv:1303.1090 [cs.SY]*, 2013.
- [43] D. Manickathu, “Comparison of ADMM formulations for MPC problems,” *ETH Master Thesis*, 2016.
- [44] J. Jerez, P. Goulart, S. Richter, G. Constantinides, E. Kerrigan, and M. Morari, “Embedded online optimization for model predictive control at megahertz rates,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3238–3251, Dec 2014.
- [45] M. Berry, M. Heath, I. Kaneko, M. Lawo, R. Plemmons, and R. Ward, “An algorithm to compute a sparse basis of the null space,” *Numerische Mathematik*, vol. 47, no. 4, pp. 483–504, 1985.
- [46] A. Topcu, *A contribution to the systematic analysis of finite element structures using the force method*. Ph.D. thesis, University of Essen, Germany., 1979.

- [47] I. Kaneko, M. Lawo, and G. Thierauf, “On computational procedures for the force method,” *International Journal for Numerical Methods in Engineering*, vol. 18, no. 10, pp. 1469–1495, 1982.
- [48] J. R. Gilbert and M. T. Heath, “Computing a sparse basis for the null space,” *SIAM Journal on Algebraic Discrete Methods*, vol. 8, no. 3, pp. 446–459, 1987.
- [49] R. Bhattacharya and G. J. Balas, “Anytime control algorithm: Model reduction approach,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 767–776, 2004.
- [50] S. Zilberstein, “Using anytime algorithms in intelligent systems,” *AI magazine*, vol. 17, no. 3, p. 73, 1996.
- [51] L. Greco, D. Fontanelli, and A. Bicchi, “Design and stability analysis for anytime control via stochastic scheduling,” *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 571–585, 2011.
- [52] V. Gupta and F. Luo, “On a control algorithm for time-varying processor availability,” *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 743–748, 2013.
- [53] Y. V. Pant, H. Abbas, K. Mohta, T. X. Nghiem, J. Devietti, and R. Mangharam, “Co-design of anytime computation and robust control,” in *2015 IEEE Real-Time Systems Symposium*, Dec 2015, pp. 43–52.
- [54] K. J. Åström and B. Bernhardsson, “Comparison of Riemann and Lebesgue sampling for first order stochastic systems,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, vol. 2, 2002, pp. 2011–2016.
- [55] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
- [56] D. P. Borgers, V. S. Dolk, and W. P. M. H. Heemels, “Riccati-based design of event-triggered controllers for linear systems with delays,” *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, June 2017.
- [57] Y. Wang, W. X. Zheng, and H. Zhang, “Dynamic event-based control of nonlinear stochastic systems,” *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, May 2017.
- [58] L. Ma, Z. Wang, and H. K. Lam, “Event-triggered mean-square consensus control for time-varying stochastic multi-agent system with sensor saturations,” *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3524–3531, July 2017.

- [59] R. Postoyan, P. Tabuada, D. Nesic, and A. Anta, “A framework for the event-triggered stabilization of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 982–996, 2015.
- [60] S. Longo, E. C. Kerrigan, and G. A. Constantinides, “A predictive control solver for low-precision data representation,” in *2013 European Control Conference (ECC)*, July 2013, pp. 3590–3595.
- [61] L. Grüne, “NmPC without terminal constraints,” in *4th IFAC Nonlinear Model Predictive Control Conference (NMPC 2012)*, vol. 4. IFAC, 2012, pp. 1–13.
- [62] H. J. Ferreau and H. Peyrl, “mpcBenchmarking Webpage,” in <https://github.com/ferreau/mpcBenchmarking>, 2015.
- [63] S. Diamond and S. Boyd, “Stochastic matrix-free equilibration,” *Journal of Optimization Theory and Applications*, pp. 1–19, 2016.
- [64] A. M. Bradley, “Algorithms for the equilibration of matrices and their application to limited-memory quasi-newton methods,” Ph.D. dissertation, Stanford University, 2010.
- [65] K. V. Ling, J. M. Maciejowski, A. Richards, and B. F. Wu, “Multiplexed model predictive control,” *Automatica*, vol. 48, no. 2, pp. 396–401, 2012.
- [66] T. V. Dang, K. V. Ling, and J. M. Maciejowski, “Banded null basis and ADMM for embedded MPC,” in *IFAC World Congress*. IFAC., 2017.
- [67] H. Ferreau, “qpoases—an open-source implementation of the online active set strategy for fast model predictive control,” in *Proceedings of the Workshop on Nonlinear Model Based Control—Software and Applications, Loughborough*, 2007, pp. 29–30.
- [68] F. Ullmann and S. Richer, “Fiordos: Code generation for first order methods.” in <http://fiordos.ethz.ch/>, 2012.
- [69] P. Zometa, M. Kögel, and R. Findeisen, “ μ ao-mpc: a free code generation tool for embedded real-time linear model predictive control,” in *American Control Conference (ACC)*, 2013. IEEE, 2013, pp. 5320–5325.
- [70] G. Banjac, B. Stellato, N. Moehle, P. Goulart, A. Bemporad, and S. Boyd, “Embedded code generation using the osqp solver,” in *56th IEEE Control and Decision*, 2017.
- [71] J. Defranza and D. Gagliardi, *Introduction to Linear Algebra with applications*. McGraw-Hill Higher Education, 2009.

- [72] T. V. Dang, K. V. Ling, and J. M. Maciejowski, “Embedded ADMM-based QP solver for MPC with polytopic constraints,” in *Proc. European Control Conference (ECC)*, 2015, pp. 3446–3451.
- [73] D. Boley, “Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs,” *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2183–2207, 2013.
- [74] A. U. Raghunathan and S. Di Cairano, “Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control,” in *Proceedings of Symposium on Mathematical Theory of Networks and Systems*, 2014, pp. 807–814.
- [75] P. Giselsson and S. Boyd, “Preconditioning in fast dual gradient methods,” in *Proceedings of the 53rd Conference on Decision and Control*, 2014.
- [76] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK users’ guide*. Siam, 1979, vol. 8.
- [77] M. A. Woodbury, “Inverting modified matrices,” *Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ*, 1950.
- [78] E. Hartley and J. Maciejowski, “Graphical FPGA design for a predictive controller with application to spacecraft rendezvous,” in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*. IEEE, 2013, pp. 1971–1976.
- [79] M. J. Koegel and R. Findeisen, “A fast gradient method for embedded linear predictive control,” in *IFAC World Congress, Milan, 2011*, pp. 1362–1367.
- [80] S. Haykin, *Communication Systems*. John Wiley & Sons, 2008.
- [81] H. K. Khalil, *Nonlinear systems*. Prentice Hall, 2002.
- [82] Y. Ji and H. J. Chizeck, “Jump linear quadratic Gaussian control: steady-state solution and testable conditions,” *Contr. Theory Adv. Tech.*, vol. 6, no. 3, pp. 289–319, 1990.
- [83] Y. Fang and K. Loparo, “Stochastic stability of jump linear systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1204–1208, 2002.
- [84] J. P. LaSalle, *The stability and control of discrete processes*. Springer Science & Business Media, 1986.
- [85] D. E. Quevedo, W. J. Ma, and V. Gupta, “Anytime control using input sequences with Markovian processor availability,” *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 515–521, Feb 2015.

Author's publications

Submitted journals

T. V. Dang, K.V. Ling and D.E. Quevedo, "Stability analysis of event-triggered any-time control with multiple control laws", IEEE Transactions on Automatic Control, DOI: 10.1109/TAC.2018.2833161.

T. V. Dang, K.V. Ling and Jan M. Maciejowski, "Faster MPC by banded null basis and ADMM". In preparation for submitting to IEEE Transaction on Control System Technology.

Conference Papers

T. V. Dang, K. V. Ling and J. M. Maciejowski, "Banded null basis and ADMM for embedded MPC," The 20th World Congress of the International Federation of Automatic Control, Toulouse, 2017, pp. 13712-13717.

K. Q. Huang, T. V. Dang, K. V. Ling and D. E. Quevedo, "Event-triggered anytime control with two controllers," 54th IEEE Conference on Decision and Control (CDC), Osaka, 2015, pp. 4157-4162.

T. V. Dang, K. V. Ling and J. M. Maciejowski, "Embedded ADMM-based QP solver for MPC with polytopic constraints," European Control Conference (ECC), Linz, 2015, pp. 3446-3451.

T. V. Dang, T. Tran and K. V. Ling, "Numerical algorithms for quadratic programming in model predictive control - An overview", Proceedings of ISSAT international conference on modelling of complex systems and environment. Danang, 2015.

T. V. Dang and K. V. Ling, "Moving horizon estimation on a chip," 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 2014, pp. 431-437.