Multi-objective design of model-predictive control and its application in missile autopilot and guidance

Vincent Bachtiar

August 2016

Department of Mechanical Engineering
The University of Melbourne

Submitted in total fulfilment of the requirements of the degree of
Doctor of Philosophy

Abstract

ADVANCED autonomous systems are typically associated with high performance requirements. A motivating example that will be the basis of the research project is a controlled defensive missile under the autopilot and guidance subsystems. In such a system, the missile is required to be able to intercept a target, which is potentially highly manoeuvrable, with precision and pace. The resulting dynamics of the engagement are non-linear and often constrained, for example, the missile must be operated within an envelope such that aerodynamic models used to develop the controller are valid. The ensuing dynamics can therefore be quite difficult to be handled by the control system.

Model-predictive control is a potentially effective method for missile autopilot and guidance due to its ability in directly handling the non-linear and constrained dynamics of a demanding missile engagement. However, model-predictive controllers typically has high levels of computational load, an issue accentuated by the fast sampling rate required in applications with fast dynamics such as missile control. Consequently, the required computational capacity of the implementation hardware – that predominantly determines the implementation cost – is an important design consideration for model-predictive control systems. Despite the motivation, current approaches mostly focus only on the optimisation of control performance and treat computational capacity as a constraint. In cases where implementation design is not known a priori, the upper-bound on required computational capacity can only be assumed, potentially leading to either a sub-optimal design that is too conservative or a design that could not meet both performance and computational requirements. This warrants the development of a multi-domain design scheme that is effective in optimising closed-loop performance as well as the required computational capacity for real-time implementation.

This thesis investigates a multi-objective design method for model-predictive control, and explores its application in advanced defensive missile control. Firstly, the mathematical formulation of the design approach is presented. The first optimised objective is the control performance metric, based on closed-loop simulations of the plant, that measures how well the controller steers the plant for a given set of initial conditions and/or tracking trajectories. Alongside

performance, the required computational capacity is to be minimised, measured by the *utilisation number* that is a multiplier metric indicating the required computational power of the implementation hardware relative to the hardware used for the closed-loop simulations performed. The tuned design parameters are the sampling time and prediction horizon length. These are associated with the controller algorithm and affect both design objectives, and denoted as the *structural* parameters. Parameters that are associated with the implementation of the controller and/or those affecting only one design objective are fixed. The solution of the optimisation problem is the *Pareto front* that defines the optimal trade-off relationship between the two design objectives. This curve provides useful insights to the practitioner and aids the design process of the model-predictive controller, for instance, the optimal performance for a range of computational capacity of the implementation hardware.

Secondly, analytical results are obtained for the proposed multi-objective optimisation problem. These include the smoothness properties of the objectives functions with respect to the parameters, particularly continuity and differentiability, as well as the bounds on the optimal parameter set. This is important as the proposed design method involves closed-loop simulations for each objective evaluation, and thus can be very time-consuming. An effective and efficient optimiser is valuable to the practicability of the proposed method. Subsequently, the obtained properties are defined in a number of theorems that are verified through numerical results and used to prescribe necessary and sufficient conditions for an effective and efficient optimiser. A compliant algorithm that satisfies the conditions is proposed and its performance justified through numerical studies.

Thirdly, the multi-objective design method is demonstrated on a proposed model-predictive integrated missile controller. Following recent developments in missile systems, the proposed controller integrates the autopilot and guidance subsystems to exploit the synergies between the two, removing in particular the lag between commanded the tracked acceleration or roll manoeuvres. The performance of the proposed model-predictive integrated missile control is numerically studied under the presence of disturbance and compared with a separated control based on proportional navigation and assumed ideal autopilot that is prevalent in industry. Numerical results show the potential superiority of the proposed controller in intercepting a target in challenging scenarios.

Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is less than 100 000 words in length, exclusive of tables, figures, bibliographies and appendices.

Vincent Bachtiar
August 2016

Acknowledgements

I would like to first of all give gratitude to Professor Chris Manzie and Dr William Moase who have provided support in an advisory role throughout my time working on this research. Their advice and guidance has added a great amount of value to this thesis, and their supervision has made my research experience both challenging and pleasurable.

The research would not have been possible without help from collaborators external to the laboratory; Tillmann Mühlpfordt, Dr Timm Faulwasser, Professor Rolf Findeisen and Dr Eric Kerrigan, who helped with publications and research direction. Tillmann, Timm and Rolf provided their expertise and helped in the first publication pertaining the thesis. In particular, Tillmann's work prior to the start of my research provided much guidance and direction for the thesis to ensure that I hit the ground running. I would like to make a deserved special mention of Eric as he provided not only his expertise for subsequent publications, but also extended advice for the direction of the project.

I would like to also thank Dr Jerome Vethecan, Andrew George, Dr Royce Smart and Dr Paul Riseborough from BAE Systems Australia. They, along with Professor Andrew Ooi and Associate Professor Nicholas Hutchins from the Fluids Mechanics Group, had their fair share of involvement in the collaborative project between BAE Systems and the Dynamics and Control Group, which subsequently made this research opportunity possible.

I was fortunate to have great support from the laboratory and department colleagues, for which there are too many to mention. I would like to particularly thank Kuan Lee who worked closely with me in the project and is never too busy to offer support. Ronny Kutadinata helped me settle down quickly and was also always available to help, both outside and inside the university environment. Changfu Zou, Jalil Sharafi, and Michael MacDonald – who was there since day one of my time in Melbourne – gave their invaluable support, both professionally and personally, and have added tremendous value to my research experience.

My parents, brother and friends, including those that have been mentioned above, have given the physical and emotional support that was instrumental for the completion of the research. It is them that have been a constant source of motivation for me in performing this research, and to whom this thesis is wholeheartedly dedicated.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

ADVANCED autonomous systems warrant multi-domain design schemes capable of meeting the challenging task of achieving the best possible system performance that is feasible under cost constraints for implementation. Such systems are becoming increasingly pervasive in many engineering technologies across many industries, including agriculture [EKSH09], manufacturing [HPT+12], transportation [APPI10] and defence [Sio04]. At the heart of the design process are the consideration of intelligent algorithms and capable implementation hardwares for the system. Although proven to be reliable, traditional approaches often treat algorithm and implementation designs separately, potentially leading to increased costs and/or sub-optimal designs that may no longer be sufficient for increasingly demanding design requirements. A multi-domain design scheme is potentially effective to improve system performance and computational resource efficiency in the design of advanced systems, not only for the physical aspect but also the associated computational platforms.

1.1 Background

Control system technologies are prevalent in defence applications, in particular where missiles are used as means of protection against the enemy [Sio04]. In recent years, with the increase of required defence capabilities in autonomous countermeasures and surveillance, advanced control systems have been increasingly used, not only for missiles, but other controlled projectiles including aerial vehicles such as drones [GKM10].

The implementation of control requires consideration across multiple domains. At the highest level, the system can be generally decomposed into the algorithm and implementation designs. The algorithm defines the control law that governs the autonomy of the controlled plant, whilst the implementation design defines the resources available for the real-time implementation of the developed control law. For the control law, model-predictive control (MPC) is an

advanced technique that promises optimal use of available hardware capacity to achieve demanding tasks. MPC is therefore suited for complex autonomous systems of high capabilities, offering much potential for applications in defence. By incorporating a dynamic model to calculate future behaviour of the controlled plant, the optimality and constraint-abiding principles of MPC allow the plant to be operated in maximal-performance regimes whilst satisfying operational limits as well as stability under disturbance.

The need for an advanced control law to achieve the best performance of an autonomous system is clear. However, such a law – most particularly in the case of MPC – typically requires costly computational capacity. Consequently, the balance between performance and cost of the required computational capacity associated with the algorithm and implementation design in MPC-based controllers is a critical consideration. This need is accentuated in defence applications where the dynamics involved are often fast, up to the order of milliseconds. In such cases, there is necessity to use computationally-capable hardware implements, which are often expensive, that can handle the high computational load of MPC.

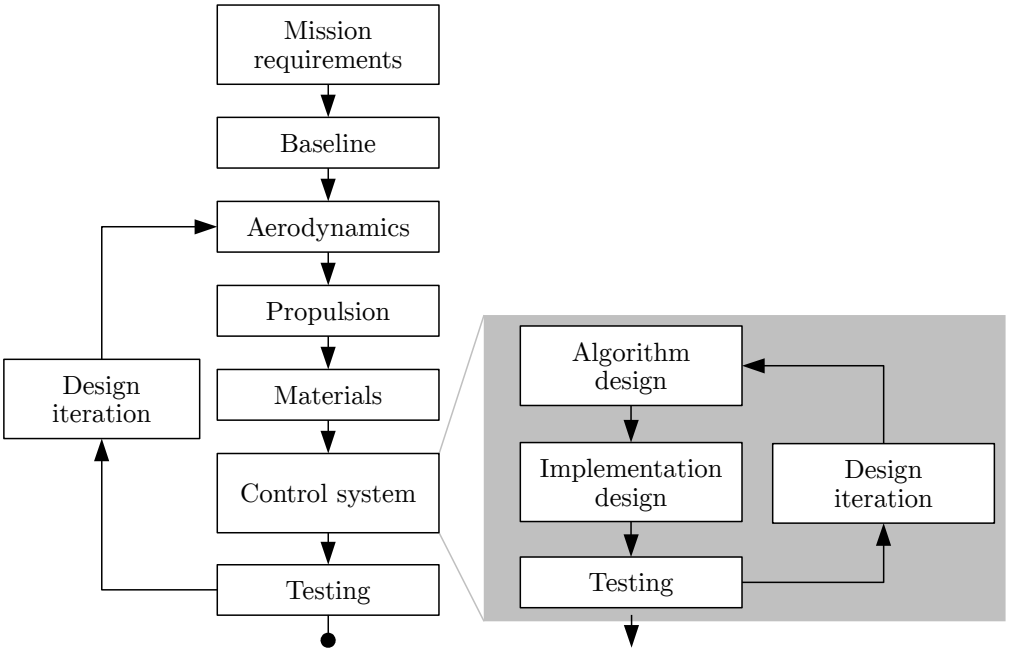

Figure 1.1: Schematic of missile system design process. Adapted from [Fle01].

In overall, the design of MPC as a digital controller has the objective of a good closed-loop performance in the presence of computational constraints that define the cost of real-time implementation. Traditionally, associated subsystems are often treated separately, each with specific design objectives that come from baseline requirements [Fle01] as illustrated in Fig. 1.1. Such an independent approach treats design objectives within individual subsystems in isolation, potentially leading to a protracted refinement cycle to meet all required thresholds. Feasibility is a potential issue whereby some requirements are unachievable due to prior design decisions on preceding subsystems. Furthermore, even if feasible, designs resulting from the separated approach can be sub-optimal, for instance, a conservative or over-specified hardware to achieve a given performance requirement. This presents a potential issue on cost that comes from extensive design iterations, further added to the cost of the required resources and implementation of the system.

1.2 Research goal

In light of the motivation above, the research project considers the design/tuning of MPC with a multi-objective design outlook and its application in an advanced missile control. The approach will consider closed-loop performance and required computational capacity as design objectives in synchrony. The two metrics are closely interrelated, often with a specifically competing relationship that can be balanced in the maximisation of performance and minimisation of the required resources. There is a potential merit in the simultaneous consideration of the two metrics as a more effective design process of MPC. In particular, such an approach will provide a selection of optimal system designs that are tailored to the given problem, providing the practitioner assistance in making well-grounded decisions and avoiding any unnecessary costs when designing the MPC.

1.3 Thesis layout

The thesis first reviews the topics of missile control, model-predictive control and multi-objective design approach in current literature. The literature review in Chapter 2 pays attention to the potential contribution of (1) a multi-objective design approach for MPC and the development of an associated optimiser that is

targeted to the problem, as well as (2) a model-predictive missile control system that can address superior specifications in performance. Subsequently, the chapter poses three research aims of the project.

The mathematical formulation of the proposed multi-objective design approach for MPC is presented in Chapter 3, formalising the design objectives and parameters that underlies the problem. The subsequent chapter analyses the proposed design scheme and presents several lemmas and theorems based on which an effective and efficient optimiser algorithm is developed on. Chapter 4 concludes by demonstrating the use of the proposed approach on two real-world test plants and validating the effectiveness and efficiency of the proposed solution method for the MOD-MPC problem. The two chapters cover the first and second research aims of the thesis.

In Chapter 5, the proposed model-predictive integrated missile control is introduced. The proposed control system is based on the integration of interrelated subsystems to achieve superior performance in engaging a manoeuvring target. To complete the fulfilment of the third research aim, Chapter 6 demonstrates the MOD-MPC approach for the design of the proposed control system.

Chapter 7 summarises the contribution of the thesis with respect to the gaps highlighted in literature review. Finally, the chapter describes potential extensions to the study for future work.

1.4 Notational conventions and definitions

For an ordered list (column vector) $\mathbf{v} \in \mathbb{R}^n$, its size is denoted by $n_v := |\mathbf{v}|$. The element values are thus $\mathbf{v} := (v_1, \ldots, v_{n_v})$. Unless stated otherwise, a set containing several ordered list is defined in calligraphy, e.g. $\mathcal{V}$ with entries $\mathcal{V} := \left\{\mathbf{v}_1, \ldots, \mathbf{v}_{|\mathcal{V}|}\right\}$. For $M \in \mathbb{R}^{n\times n}$, $\mathbf{v} \in \mathbb{R}^{n\times m}$ and $s \in \mathbb{R}$ with appropriate n and m, $\|\mathbf{v}\|_M^2 := \mathbf{v}^\mathsf{T} M\mathbf{v}$ and $\|\mathbf{v}\| := \sqrt{v_1^2 + \ldots + v_{n_v}^2}$.

'm-min' denotes a multi-objective optimisation (minimisation) problem that finds the Pareto optimal (see Definition 1.5) set of points given a search space/set to optimise in, and distinguished from 'min' that denotes a single-objective problem. $\mathsf{U}[a, b]$ is a random number distributed uniformly between a and b. Binary operators such as $=, <, \leq$, etc. define element-wise relationship. The big-$\mathcal{O}$ nota-

tion is such that $f(x) = \mathcal{O}(g(x)) \Leftrightarrow \exists a \in \mathbb{R}_{>0},\ x_0 \in \mathbb{R}_{>0} : |f(x)| \leq a|g(x)|\ \forall x \in [0, x_0)$. Let $\mathcal{O}(g_1(\cdot), \ldots, g_n(\cdot)) = \sum_{i=1}^{n} \mathcal{O}(g_i(\cdot))$.

Definition 1.1 (Continuity [Ste11])**.** $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is continuous at $s \in \mathbb{R}^n$ if $\lim_{x \to s} f(x) = f(s)$.

Definition 1.2 (Differentiability [Ste11])**.** $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $s \in \mathbb{R}^n$ if it can be approximated linearly in the neighbourhood of the point, i.e.

$$f(s + \delta) = f(s) + \Delta(s) \cdot \delta + L(s, \delta) \cdot \delta,$$

for some gradient $\Delta(s) \in \mathbb{R}^n$ that is independent of δ. $\lim_{\delta \to 0}\ L(s, \delta) = 0, \forall s \in \mathbb{R}^n$ constitutes higher order terms. Differentiability at s implies continuity at s.

Definition 1.3 (Lipschitz continuity [Kha02])**.** A function $(x(t), u(t)) \mapsto f(x(t), u(t))$ is globally Lipschitz continuous in x, uniformly in u and t if

$$\|f(x(t), u(t)) - f(y(t), u(t))\| \leq L\|x(t) - y(t)\|$$

for all $x \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$, and $\forall u \in \mathbb{R}^{n_u}$.

Definition 1.4 (Domination [Deb01])**.** For a vector containing several objective values $\boldsymbol{\ell}(\cdot)$, $\boldsymbol{\ell}(\mathbf{p}_\bullet) \prec \boldsymbol{\ell}(\mathbf{p})$ or $\mathbf{p}_\bullet \prec \mathbf{p}$ denotes that the point $\boldsymbol{\ell}(\mathbf{p}_\bullet)$ dominates the point $\boldsymbol{\ell}(\mathbf{p})$. This is true iff $\ell_i(\mathbf{p}_\bullet) \leq \ell_i(\mathbf{p})$ for all $i \in \{1, \ldots, n_\ell\}$ and $\ell_i(\mathbf{p}_\bullet) < \ell_i(\mathbf{p})$ for at least one i.

Definition 1.5 (Pareto optimality [Deb01])**.** A point $\boldsymbol{\ell}(\mathbf{p}_\bullet)$ with $\mathbf{p}_\bullet \in \mathcal{P}$ is a Pareto point iff there does not exist another design choice $\mathbf{p} \in \mathcal{P}$ such that $\boldsymbol{\ell}(\mathbf{p})$ dominates it.

Definition 1.6 (Competing functions)**.** Two functions $s \mapsto f(s)$ and $s \mapsto g(s)$ are competing with each other in the design set $\mathcal{S}$ iff $s \mapsto f(s)$ is monotonically increasing and $s \mapsto g(s)$ is monotonically decreasing, or vice versa, for all $s \in \mathcal{S}$.

List of notation

Variables, functions and constants

$\mathbf{0}_{n \times m}$	$n \times m$ matrix with zero elements
$\mathbf{1}_{n \times m}$	$n \times m$ matrix with 1 as all elements
a	polynomial constants in solution time upper-bound model

h	sampling time
ℓ	design objective vector
o	predictive state vector after disturbance
$\mathbf{p}$	design parameter vector
u, u	input vector in continuous-time (of plant and prediction, respectively)
$\mathbf{u}$	predictive input vector in discrete-time
w	disturbance vector
x, x	continuous state vector (of plant and prediction, respectively)
$\mathbf{x}$	predictive state vector in discrete-time
$\mathbf{z}$	concatenated optimisation variable
A	state transition matrix
$\mathcal{A}$	active set
B	input transition matrix
I_n	$n \times n$ identity matrix
J	open-loop cost function
N	number of prediction horizon steps
$\mathcal{P}$	design parameter set
Q	state cost weight
R	input cost weight
S	state-input cross cost weight
T	length of prediction horizon
U	closed-loop cost function
$\mathcal{U}$	predictive input (constraint) set
V	sum of weighted closed-loop cost functions
$\mathcal{X}$	predictive state (constraint) set
η	utilisation (capacity) number
μ	Lagrange multiplier vector
ψ	optimiser convergence metric
σ	standard deviation
ζ	solution time upper-bound model
ζ_{FLOP}	required floating-point operations

Missile variables

$\mathbf{a}$	relative acceleration

$\mathbf{a}_\mathrm{M}$, $\mathbf{a}_\mathrm{T}$	acceleration (of missile and target, respectively)
q	pitch rate
r	relative distance (range)
$\mathbf{s}$	relative position
$\mathbf{s}_\mathrm{M}$, $\mathbf{s}_\mathrm{T}$	position (of missile and target, respectively)
$\mathbf{v}$	relative velocity
$\mathbf{v}_\mathrm{M}$, $\mathbf{v}_\mathrm{T}$	velocity (of missile and target, respectively)
C^L	aerodynamic lift coefficient
C^M	aerodynamic pitching moment coefficient
F_L	aerodynamic lift
M	aerodynamic pitching moment
V_M	missile speed
α	angle-of-attack
χ	seeker angle
δ	effective actuation (e.g. fin deflection angle)
δ_c	commanded fin deflection angle
Δ	miss distance
λ	line-of-sight angle
θ	pitch angle

Operators

$\otimes$	element-wise multiplication
$\oslash$	element-wise division
$\partial_s(\cdot)$	derivative evaluated at s, e.g. $\partial_s(M) := \left.\frac{\partial M}{\partial s}\right\|_s$
$\Im(\cdot)$	imaginary part
$\lambda_i(\cdot)$	i^{th} eigenvalue

Subscripts, superscripts and embellishments

$(\cdot)_\bullet$	Pareto optimal
$(\cdot)_\mathrm{c}$	related to entirely continuous design parameters
$(\cdot)_\mathrm{o}$	steady state value
$(\cdot)_\mathrm{s}$	related to the searched set
$(\cdot)_\mathrm{t}$	related to terminal state
$(\cdot)_\mathrm{z}$	related to the dense OCP representation

$(\cdot)_{\mathrm{C}}$	related to the competitive set
$(\cdot)_{\mathrm{F}}$	related to feasibility
$(\cdot)_{\mathrm{S}}$	related to stability
$(\cdot)_{\mathrm{SQP}}$	related to the sequential quadratic programming algorithm
$(\cdot)^{+}$	value after perturbation, e.g. $M^{+} := M(s + \delta_s)$
$(\cdot)^{*}$	optimal value
$(\cdot)^{\mathrm{CL}}$	related to the closed-loop system
$(\cdot)^{\mathrm{OL}}$	related to the open-loop system
$\widetilde{(\cdot)}$	related to tracking
$\overline{(\cdot)}$	upper-bound, or related to discrete-time formulation, or mean value
$\underline{(\cdot)}$	lower-bound

Chapter 2
Literature review

MULTI-OBJECTIVE design of model-predictive control and its application in a missile system underpin the aims of this research. This chapter provides an overview of the current literature on the topic in hand. Relevant studies are drawn out to elaborate the value of the thesis for the motivating background in Chapter 1. The chapter first introduces the basics of missile control and its different components. Then, model-predictive control (MPC) is introduced and its potential value in missile control applications is highlighted. Finally, studies in multi-objective optimisation are reviewed to found the multi-domain design approach that is proposed in this research. Literature gaps are presented along the way, which are subsequently summarised to emphasise the potential contribution of this work, as the basis to finally propose the aims of the research.

2.1 Model-predictive control

Model-predictive control is a model-based, optimal control technique that has the capabilities to directly handle non-linearities and constraints whilst operating near optimal conditions and guaranteeing stability even under disturbance. Since its introduction in the 1960s, MPC has grown into arguably the most promising control architecture to replace classical PID controllers. It is the only advanced control architecture to have a significant impact in the industrial control engineering [Mac02]. Its main advantage in the ability to optimally handle multi-variable plants and constraints yields a high-performance controller that operates plants at their maximum capabilities [CB04, GPM89].

In most applications, MPC is implemented in discrete-time such that control actions are applied at intervals for some sampling time h. The idea of MPC revolves around the use of a prediction model – an explicit mathematical model representing the dynamics of the controlled plant – in an optimisation setting to predict future plant behaviour in response to changes in the control input(s) over a finite prediction horizon T. The prediction horizon length is typically an integer multiple of sampling time, defined by N prediction steps, so that $T = Nh$. For a

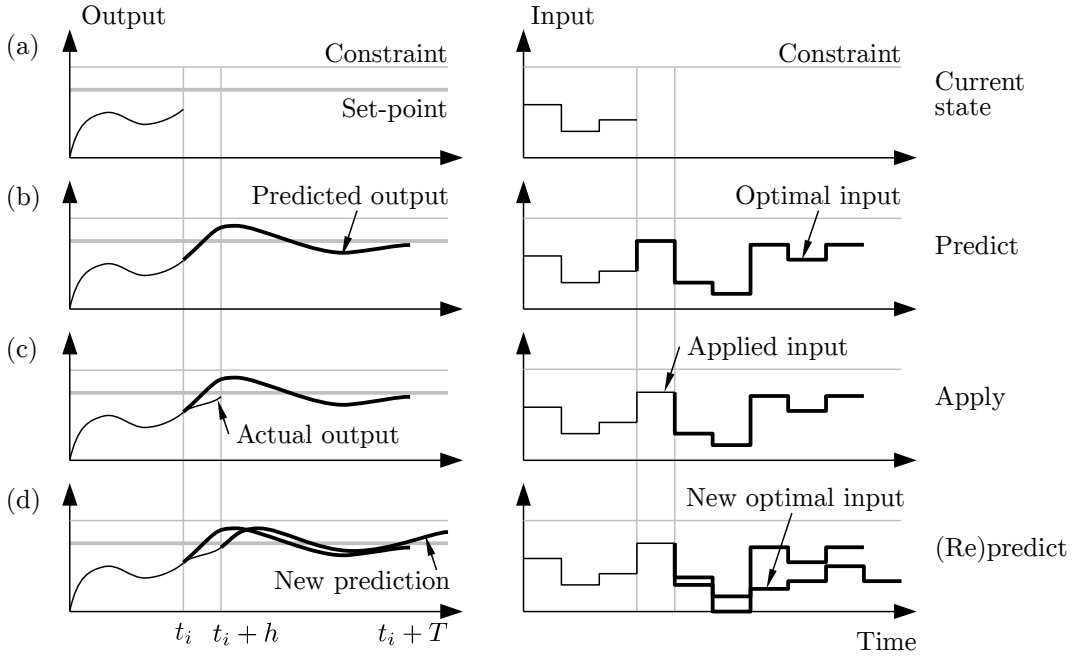

Figure 2.1: Illustration of the receding horizon in MPC. Adapted from [She12].

discrete-time configuration, the basic operation of MPC is illustrated in Fig. 2.1. In subfigure (a), the controlled plant is at its current state at some sampling instant t_i. At the sampling instant, and any subsequent ones for that matter, MPC solves an optimal control problem (OCP) whose solution represents the control input sequence that optimises a specified cost function measuring control performance, subject to the prediction model and the modelled operating constraints, as illustrated in subfigure (b). The first portion of the optimal control solution is applied to the plant, and the system evolves as per subfigure (c). Note that the predicted and actual behaviour of the plant does not necessarily match. Finally, in subfigure (d), the process is repeated at the next sampling instant with the new (current) state of the plant and the receded prediction horizon.

A basic form of the OCP might look as the following

$$(\mathbf{x}^*(\mathbf{p}), \mathbf{u}^*(\mathbf{p})) := \arg\min_{(\mathbf{x},\mathbf{u})} \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}),\ \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) := \sum_{k=0}^{N-1} \left\| \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}} + \|x_N\|^2_{Q_t} \tag{2.1a}$$

$$\text{s.t. } x_0 = \mathsf{x}_i \tag{2.1b}$$

$$x_{k+1} = \overline{f}(x_k, u_k, \mathbf{p}) \coloneqq x_k + \int_{t_i}^{t_i+h} f(x(\tau), u_k, \mathbf{p})\, d\tau \quad \forall k \in \{0, \dots, N-1\} \tag{2.1c}$$

$$x_k \in [\underline{x}, \overline{x}]\,,\ u_k \in [\underline{u}, \overline{u}] \qquad \forall k \in \{0, \dots, N-1\} \tag{2.1d}$$

The minimised cost function in (2.1a) is the chosen metric representing the open-loop performance of the plant within some prediction horizon N. This cost is commonly composed of the *stage cost* defined by the summation term and *terminal cost* defined by the second term weighted by Q_t. The cost function is to be optimised subject to (2.1b) and (2.1c) that define the prediction model used in the OCP, whilst the constraints (2.1d) define the operational constraint that the predicted states and inputs must satisfy. In this form, the OCP is defined in discrete-time with predicted states $x_k, \forall k \in \{0, \dots, N\}$ and inputs $u_k, \forall k \in \{0, \dots, N-1\}$.

The solution of the OCP $(\mathbf{x}^*(\mathbf{p}), \mathbf{u}^*(\mathbf{p}))$ is the predicted optimal response of the plant within the prediction horizon. This depends on some design parameters $\mathbf{p}$ of the OCP, that can include attributes of the cost function such as the cost weights $\overline{Q}$ and $\overline{R}$, as well as sampling time h and prediction horizon length N. The open-loop value function is the value of the cost function at the optimal solution, defined as $\overline{J}^* \coloneqq \overline{J}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{p})$.

Non-linearities in the plant dynamics can be handled in various ways. Common examples include modelling to directly capture the complexities in the prediction model used in the OCP, as well as the constraining the predicted states and inputs so that the plant operates near where the simplified (e.g. linearised) prediction model is valid. In doing so, the control law calculates input commands with better information of the plant behaviour than most traditional controllers. Furthermore, the ability of MPC in incorporating and handling constraints in the prediction allows for the plant to be operated near its limits. This results in a superior performance to more traditional control methods, whereby ad-hoc approaches are used to handle constraints, for instance, operating points that are at a safe distance away from the constraints are often chosen to avoid constraint violation, albeit being sub-optimal [Mac02]. The optimal nature of MPC enables more effective operation near the constraint boundaries without violating them. The ensuing control law, however, is non-linear and is associated with more involved analytical solutions.

With a finite prediction horizon, the general form of MPC does not guarantee stability [BWG91]. In the early developments, most of the theories derived to guarantee stability in MPC revolve around the use of a *terminal constraint* as an additional constraint to the OCP that imposes the terminal state x_N to be within some predetermined terminal set, for instance, the origin as the terminal point [MM91, MM90]. Subsequent developments use the notion of a terminal set to increase the feasibility of the optimisation problem and allow for shorter prediction horizons, alleviating some of the computational cost of the controller [MR09, MRRS00, MM93]. To make the problem formulation simple for computational purposes, the terminal set has been approximated as an ellipsoid [CA98] and a polytope [CDK03]. In lieu of using a terminal constraint, an appropriate terminal cost and a sufficiently long prediction horizon can be used to guarantee stability for a given set-point [KM00, PZ95]. This may be more relevant in applications where computational cost needs to be kept minimal, prohibiting the use of the additional terminal constraint.

The main drawback in MPC is the high computational load as it needs to solve the associated optimisation problem in each sampling instant. First, the OCP is a constrained optimisation problem so that an explicit form of the solution is not directly obtainable. A numerical solver is required to solve the problem, and thus the time needed by the solver to find the control action that is to be applied is a fundamental factor in MPC. Furthermore, applications in systems involving difficult dynamics often require the use of a non-linear prediction model to minimise/eliminate plant-model mismatch to avoid detrimental effects such as steady-state output error or instability. With a non-linear prediction model, the OCP becomes a non-convex optimisation which is more difficult to solve than a convex optimisation associated with a linear prediction model. This presents a challenge especially for applications in electromechanical plants requiring fast sampling times that heavily limit the time to solve the optimisation problem, including advanced defensive missiles.

2.2 Missile control

The basis of missile control is the manipulation of actuators to produce aerodynamic forces and moments that steers the missile to intercept a target. This is

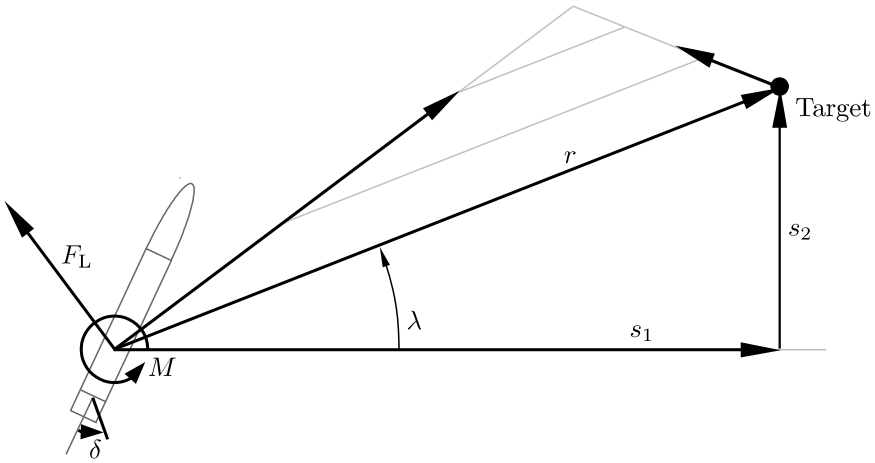

Figure 2.2: Illustration of a missile-target engagement.

illustrated in Fig. 2.2. In the figure, the missile is controlled in a two-dimensional plane for one of the control channels, e.g. pitch control, with the effective actuation δ that generates aerodynamic lift and pitching moment F_L and M, respectively. The missile is a distance r away from a moving target, with line-of-sight (LOS) angle λ. Intercept is achieved when the locations of the target and missile coincides or are sufficiently close to each other so that the range of the warhead explosive device can reach the target.

Component-wise, missile control can be sub-divided into several distinct subsystems as illustrated by a simplified diagram in Fig. 2.3. Given a target velocity and position relative to the missile, the guidance law calculates the required acceleration for a successful intercept. The information of the target is measured by a seeker/homing system that is either internal or external to the missile, or both. The autopilot takes acceleration commands from the guidance law and generates actuation commands in an attempt to track the desired accelerations. The inertial measurement unit (IMU), which includes accelerometers and gyroscopes, measures the attitude (pitch, bank and yaw), translational acceleration and angular rate of the missile and feeds this information back to the autopilot. Such measurements are not obtained instantly, and can be modelled with a delay, e.g. a second-order system [Jac10]. The degree of accuracy of the measurements is subject to disturbance that can be handled by appropriate countermeasures [TPB00].

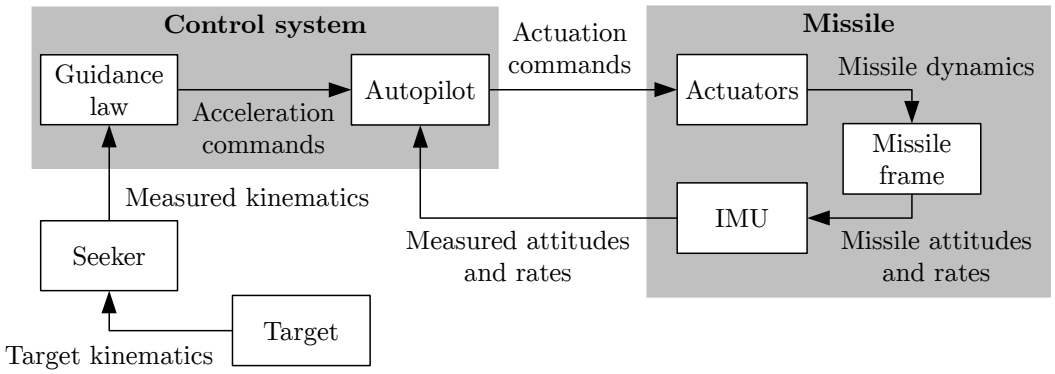

Figure 2.3: Schematic of missile feedback control.

2.2.1 Guidance

For guidance, proportional navigation (PN) was devised in the mid 1900s [Adl56] and is based on parallel navigation that relies on the line-of-sight (LOS) direction. The LOS is defined by the elevation and azimuth angles of the target as viewed from the missile. The idea is that the rate of line-of-sight (LOS) angle λ must be zero i.e. $\dot{\lambda} = 0$ for an intercept, as demonstrated in Fig. 2.2. PN achieves this by commanding a missile acceleration proportional to the LOS rate [PBL10a]

$$a_{\mathrm{c}} = K v_{\mathrm{cl}} \dot{\lambda} \tag{2.2}$$

where a_{c} is the commanded missile acceleration, v_{cl} is the closing velocity between the missile and target, and K is a control gain. Given a positive closing velocity so that the missile is approaching the target, a zero LOS rate guarantees intercept. However, the zero LOS rate condition in itself is neither a sufficient nor necessary condition for intercept. The missile is generally required to travel faster than the target, although this depends on the relative position and velocity between the missile and the target. Ever since its first successful application, PN is the most widely used guidance law as noted by [NZ81] and others [Yan08,Zar12].

PN guidance laws can be categorised based on the direction of the commanded acceleration. True proportional navigation (TPN) calculates desired acceleration commands normal to the LOS, and pure proportional navigation (PPN) calculates the acceleration commands normal to the missile velocity. In planar engagement, PPN has been shown to be superior to TPN in terms of implementation, robustness and control effort required [SM90], as well as

capture region [Gue76]. A law dubbed ideal proportional navigation (IPN) commands acceleration normal to the relative velocity between the missile and the target. This approach has been shown to have a larger capture region whilst requiring more control effort [YC92]. The principles of PN has also been extended into other guidance laws, the most popular being augmented proportional navigation (APN) which uses the acceleration of the target as additional information to the LOS rate in calculating the desired missile acceleration [Yan08].

More modern guidance laws are based on optimal control principles. These guidance laws minimise the required missile acceleration, i.e. the controller effort, in an engagement [PBL10b]. Under the assumptions of zero missile dynamics (ideal autopilot), a non-accelerating target, as well as a constant closing velocity $v_{\text{cl}} := \dot{r}$ so that $r = v_{\text{cl}} t_{\text{go}}$, the optimal guidance law simplifies down to

$$a_{\text{c}} = 3\left(\frac{s_2 + \dot{s}_2 t_{\text{go}}}{t_{\text{go}}^2}\right) = 3v_{\text{cl}}\left(\frac{s_2 v_{\text{cl}} + \dot{s}_2 r}{r^2}\right).$$

For small values of λ so that $\lambda \simeq s_2/r$ and $\dot{\lambda} \simeq (s_2 v_{\text{cl}} + \dot{s}_2 r)/r^2$, the expression above yields the PN law (2.2). This concludes that, with a gain of $K = 3$, the PN law is equivalent to an optimal guidance law. These types of missile guidance laws have been shown to reduce the terminal miss distance and acceleration command saturation compared to PN [Cot71]. A minimised commanded acceleration (control effort) is desirable as missile acceleration capabilities are often limited. Although a greater acceleration capability is generally desirable for a missile, there are cases where an excessive acceleration can lead to larger miss distance due to overshoot [Zar99]. This can happen due to an image re-solution of the target when the missile is close enough to the target and the tracking point instantaneously shifts from the centroid of the target to a specific part of the target, e.g. the warhead of an attacking missile.

The popularity of PN is a testament to its performance. However, as an optimal guidance law, PN predicts future behaviour until intercept, essentially relying on the time-to-go t_{go} that measures the time until the missile intercepts the target. Optimal guidance laws with a (shorter) receding prediction horizon that is fixed irrespective of the time-to-go have been proposed in [KKK00, KYK01] as an attempt to move away from the reliance on time-to-go that is only available as an

estimate. However, these approaches are not based on parallel navigation (LOS rate regulation), but on nullifying the separation between the missile and the target, thus susceptible to large overshoots especially in cases of non-zero altitude angles which were not considered by the authors.

2.2.2 Autopilot

In autopilot applications, missile dynamics are typically decoupled into separate channels of pitch, yaw and roll. There are several examples of studies for the individual pitch [CBG03, KKS04, NRR93, Rei92, SC93], yaw [BM93] and roll control [HC07, TGKP11] in the literature. With this decoupled configuration, a missile can either be controlled in a sequential bank-to-turn (BTT or *twist-and-steer*) or skid-to-turn (STT) strategy. For the former, the missile manoeuvres by making a banked turn that is first achieved by rolling to a desired direction then pitching to turn whilst regulating sideslip angle to zero. The latter strategy involves control in the yaw plane to make a sideslip angle to turn, before pitching accordingly. For an axisymmetric missile, the STT strategy is preferred as it is quicker than a coordinated banked turn. However, for a missile that is asymmetric and has better manoeuvrability in either pitching or yawing, a banked turn can take advantage by turning via the appropriate channel [Sch94].

Autopilot controls the actuators of the missile to track the acceleration command given by the guidance law. The type of actuators varies, including control surfaces (fins) as well as active propulsion. Any actuator configuration can be aggregated into three control variables that define the effective actuation input for roll, pitch and yaw through the use of a mixing logic. An example for an STT missile with a cruciform fin configuration is given in [Sio04] and one for a BTT missile is given in [Sch94]. Actuation does not occur instantaneously and their response can be modelled by a first-order [KKS04, Sch94] or a second-order response [DHS01, NRR93, SC93], often with saturation.

High-performing missiles are often pushed to operate near their physical limits, at high angles of attack where aerodynamics are highly non-linear [GQD12]. Therefore it is crucial for the autopilot to directly take into account plant non-linearities and handle constraints explicitly, to maximise missile performance without exceeding operational limits. Current challenges in missile control often

revolve around the inability to account for the non-linearities, changes in missile behaviour during flight and different missile configurations [Jac10].

Early missile autopilots are based on proportional control. A 'three-loop' design was employed in [NN84] involving a corrector for angular rate in the inner loop and acceleration in the outside. Other authors looked at the use of PI control [FSF97], gain-scheduling [NRR93] and sliding-mode control [TI98]. Although effective, these classical controllers are unable to handle constraints and the highly non-linear dynamics of the missile directly, potentially resulting in a sub-optimal performance of the missile.

2.2.3 Integrated autopilot and guidance

Traditionally, guidance law and autopilot have normally been designed separately. Much effort has been concentrated on integrating the autopilot and guidance subsystems to improve missile performance [MO01a, MSOM04]. The unification of the control algorithm has direct benefits for the performance of the missile, primarily by circumventing the lag between the commanded and tracked acceleration from guidance and autopilot, respectively. Several control strategies have been proposed following this design outlook, including sliding mode control [ISG07, SIG06, ST09], state feedback regulators [MSO03, PJ99] and model-based approaches [KWK16]. The approach used in [MSO03] involves linearisation of the dynamic missile model with state-dependent coefficients, and stabilisation via the solution of the state-dependent Riccati equation. Similar results are obtained in [SIG06] with an integrated sliding-mode control algorithm. The improved performance is achieved by exploiting the synergies between the two subsystems [MO01b]. The integration of subsystems can also simplify the controller design process, possibly alleviating associated costs.

2.3 MPC in missile control

The dynamics of the missile are highly non-linear. Coupled with the non-linearities, models for the stability and control derivatives used are often accurate only locally around an operating point. Therefore, the model of the missile is constrained within a specific set of operational bounds, as well as constraints related to the physical limits of the missile frame and actuators.

Lastly, it goes without saying that mission requirements often demand high-performing missiles that can operate near constraints. Given these underlying aspects, an MPC-based controller naturally fits in the problem setting due to its features in handling non-linearities and constraints.

Naturally, there have been attempts to address the performance requirements in missile control with model-based approaches in the literature. For instance, the prevalent proportional navigation law for guidance can be derived from an unconstrained model-based control with a linear prediction model and a prediction horizon based on the engagement period [PBL10b]. In autopilot, a generalised predictive control has been applied for pitch autopilot [CBG03]. The approach uses an analytical formulation to obtain control commands rather than an online optimisation, hence reducing the computational load significantly. However, with this approach, constraints can only be imposed on the derivative of the control variable. For general flight control, MPC has been proposed as a theoretically fault-tolerant scheme owing to the reconfigurability of its prediction model and ability to handle constraints explicitly, enabling it to control airframes under damages such as actuator failures [KC02, Rui04]. More recently, a stable non-linear MPC approach with full state and input constraints has been studied, albeit for the case of roll autopilot [HC07]. Roll dynamics are simpler than that of pitch or yaw, therefore less burdensome to calculate in the ensuing OCP, allowing for a more straightforward implementation of MPC. Studies of MPC for the more general non-linear and constrained control of missile, particularly in the pitch and/or yaw axis, remain scarce.

The consideration of non-linear dynamics and/or constraints in MPC offer much potential for applications in high-performing missiles. This, however, comes at a cost of an increased computational load from the need to solve an optimisation problem at each sampling instant. This issue is accentuated in missile control applications where high sampling rates are needed. Typical rates used in the studies of missile control lie between 80 to 200 Hz [BTW12, HC07]. This corresponds to sampling times of around 15 to 5 ms, which are much faster than the sampling times in the process industry where MPC is traditionally used. Nonetheless, with the advancement in computing processors, applications of MPC have grown to be more widely used in industries involving relatively faster dynamics such as transportation [AAT08, ATA09].

The issue of limited computational time of MPC in missile control has been discussed and mitigated to some extent, often with the compromise in performance. For an application in missile pitch control, some studies tackle the problem of the OCP being unsolved within the sampling time by using a sub-optimal feedback based on the optimal input sequence from the previous sampling instant whilst preserving stability [HC07]. Another approach is to use a reduced form of MPC to lessen the load of the online optimisation. One study proposes a predictive functional control scheme, formulating the input sequence as a weighted linear sum of basis functions to reduce the number of decision variables of the OCP and hence computational load [TC12]. However, a linear prediction model is used and state and output constraints are removed to obtain an analytical optimal solution. Due to the simplification of constraints, similar to the previously cited general predictive control [CBG03], the approach would suffer the same drawbacks in control performance.

2.4 Calibration of MPC

The original full form of MPC can be calibrated, i.e. designed/tuned, to achieve a desired design. This can be done in lieu or as an addition to the use of reduced or sub-optimal designs to minimise the computational load of MPC. Calibration involves the tuning of pertinent parameters of the controller, in particular those associated with the formulated OCP (2.1). Some of the parameters of the OCP that can be calibrated include attributes of the cost function such as the cost weights $\overline{Q}$ and $\overline{R}$, as well as sampling time h and prediction horizon length N. These parameters can be calibrated to optimise the design objectives of MPC.

2.4.1 Optimisation of performance

Control performance, especially in closed-loop, is first and foremost the design metric that is to be maximised when designing MPC, or any controller for that matter. This follows on arguably the first attempt in controller calibration in the Ziegler-Nichols gain tuning for proportional controllers [ZN42]. Along with performance, implementation cost is another optimisable metric, that is predominantly dictated by the computational capacity required to functionally implement the controller. Although control performance is traditionally the more preva-

lently considered design objective, required computational capacity is an equally important design consideration, especially for model-based controllers which are typically power-hungry and expensive as indicated earlier in the introduction of MPC. The influence of computational cost is further amplified in control of plants with difficult dynamics where fast sampling rates necessitate a high processing power, for instance in missile control.

Much focus has been given to find the best control performance in single-objective optimisation, separate to the consideration of the required computational capacity. Following on the calibration of proportional controllers, performance optimisation is investigated for more complex controllers such as sliding mode [Ha96, RP01]. Calibration of model-based controllers soon follows, but there are still a number of knowledge gaps in existing design approaches. MPC tuning for control performance is mostly done via methods that rely on rules-of-thumb and general guidelines [GS10, QB03, RU97]. Further developments have been made consequently, using metaheuristic optimisers such as particle swarm optimisation [JMK14] and genetic algorithms [vdLSY08], as well as gradient descent [BFFB12], for the single-objective optimisation of MPC performance.

Several multi-objective optimisation techniques for control system design have also been studied for the optimisation of control performance. In these studies, more than one metric for performance are considered, including, for example, the overshoot and rise-time of the plant output. Similar to that of the single-objective counterpart, metaheuristic methods are prevalently used for the multi-objective tuning of classical control, such as PID [AdSC12, RMGNSB13, XLG10], sliding mode control [MTB14, TMTCV14], as well as others [RMBSM14]. A similar approach is applied in MPC tuning by using an off-the-shelf method of goal attainment [ET10, VFT08]. Although more systematic than general tuning guidelines, these techniques provide non-specialised means that do not exploit certain characteristics of the problem and potentially require a rather exhaustive and possibly computationally impractical search to produce the optimal design set. As an alternative to methods based on guidelines and metaheuristics, analytical approaches employing problem simplifications have been proposed [BKS14, SE11, SC98]. However, these typically overlook some aspects of the original problem such as explicit constraint handling.

Lastly, control performance is often represented by the open-loop value function ($\bar{J}^*$ from the example OCP (2.1)) [BKS14, BB14]. Since the controller is implemented in closed-loop, the open-loop value function might not be the best measure for control performance, albeit still a good indicative quantity. Instead, a measure based on the closed-loop simulation of the plant can be used [MHL99].

2.4.2 Multi-objective optimisation of performance and cost

The studies discussed so far consider control performance as the sole design objective, whether with a single- or multi-objective outlook, for instance, trading-off rise-time and overshoot. The approach separates algorithm and implementation design, paying focus on only the former and reveals only half the insight in control design. Implementation design largely determines the cost for the controller hardware and is often not known a priori, thus is a part of the overall design process. The co-design of algorithm and implementation provides a more comprehensive technique that optimises control performance as well as implementation cost that is dictated by the required computational capacity for real-time implementation. Rather than treating the required capacity as a fixed constraint, it should be co-optimised alongside control performance, avoiding system over-design or the need to re-design the system.

In consideration of the above discussion, the value of a co-design approach in streamlining the design process of control systems has been noted [AH14]. Further, the fundamental concept of an algorithm and implementation co-design for real-time optimisation has been studied [Ker14, SLKC13], although analytical results to support applications in MPC are still yet to be fully developed. In particular, investigations involving required computational capacity as a design consideration alongside control performance are limited. Furthermore, previous studies, such as [BKS14, ET10, JMK14], have typically only considered parameters that only affect the closed-loop performance, primarily the cost function attributes of the OCP, since they only aim to optimise performance. Parameters that affect both performance and implementation cost – such as sampling rate and prediction horizon length – are excluded from the analysis and treated as fixed. Nonetheless, these *structural* parameters have an underlying role for MPC design improvement, particularly due to the fact that they prescribe not only control performance but also computational attributes.

2.5 Multi-objective optimisation

The two design metrics of performance and required computational capacity (implementation cost) are interrelated. In particular, they are often competing, so that an improvement on one metric must likely come at the expense of the other. Therefore, there is an underlying trade-off relationship between the two. On one extreme, a controller can be designed with, for example, fast sampling rates and high-fidelity models so that it can best predict plant dynamics to optimally control the plant. At this extreme, the computational load of the controller will be at the highest levels. On the other hand, controllers can be designed with shorter prediction horizon lengths or a reduced form to alleviate computational load at the expense of a reduced performance.

The balance between control performance and required computational capacity is a critical consideration in MPC design. This idea is encapsulated in a multi-objective optimisation problem, as formulated in the following

$$\begin{aligned} \mathcal{P}_\bullet(\mathcal{P}_\mathrm{s}) \coloneqq \ & \arg\,\mathrm{m}\text{-}\min_{\mathbf{p}} \ \boldsymbol{\ell}(\mathbf{p}) \\ \text{s.t.} \ \ & \mathbf{p} \in \mathcal{P}_\mathrm{s}. \end{aligned}$$

The multi-objective optimisation problem searches the design parameter set $\mathcal{P}_\mathrm{s}$ for the optimisation (minimisation) of the design objectives contained in the objective vector $\boldsymbol{\ell}$. The solution is the optimal design set is denoted by $\mathcal{P}_\bullet$, namely the *Pareto* optimal solution. In the objective space, the solution is the Pareto front $\mathcal{L}(\mathcal{P}_\mathrm{s}) \coloneqq \{\boldsymbol{\ell}(\mathbf{p}),\, \forall \mathbf{p} \in \mathcal{P}_\bullet(\mathcal{P}_\mathrm{s})\}$, at which one objective cannot be improved without compromise of all the other objectives. The curve[1] $\mathcal{L}$ defines the optimal trade-off between the objectives to assist with the design decisions for the optimisation of the design objectives $\boldsymbol{\ell}$, which in this case are the control performance and required computational capacity, as illustrated in Fig. 2.4. For instance, the Pareto curve indicates the best performance for a range of computational capacity, as well as the sensitivity between the two objectives, for example, the required cost in computational capacity for a unit improvement in closed-loop performance.

[1]The term *curve* is used as opposed to the more general term of *manifold* as focus is given on trade-offs between two objectives.

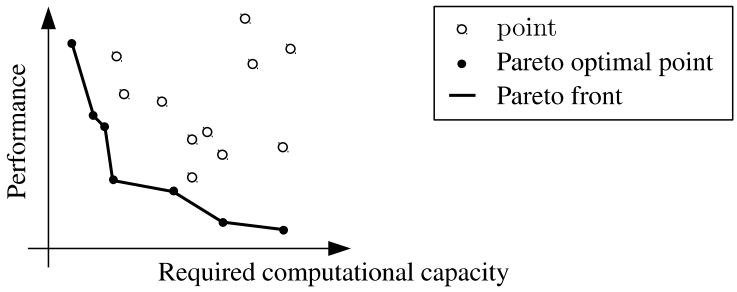

Figure 2.4: Illustration of the Pareto front in a set of objective points.

Both objectives depend on a number of design parameters of the controller, concatenated together in $\mathbf{p}$. With the outlook of multi-objective optimisation, focus will be on the design parameters that affect both objectives. These are the structural design parameters that include, but are not limited to, the sampling time and prediction horizon length. The relationship between the objectives and parameters is not always straightforward. In the case where state and input constraints are absent from the OCP, the closed-loop control performance have been shown to be non-monotonic with respect to the sampling time [LSA71], particularly so near the limit of zero sampling time [MK71]. The effect on sampling time on control performance has been studied further in the topic of real-time optimisation [BC08, BDNB08], albeit for the unconstrained problem.

Other than the non-trivial relationship between objective and performance, the evaluation of a point $\ell(\mathbf{p})$ involves a set of closed-loop simulations of the controlled plant which can be very time-consuming. Multiple simulations are necessary to investigate, for example, a range of initial conditions or average performance under disturbance. Therefore, the optimiser used needs to be accurate and fast in finding the optimal solution. Examples of multi-objective optimisers include those based on derivative-free approaches such as evolutionary algorithms [DPAM02], line search [CMVV11], particle-swarm optimisation [CCL02], simulated annealing [BSMD08] and Lipschitzian optimisation [WIHM08]. Gradient-based optimisers include those of gradient-descent [BFFB12] as well as more involved approaches such as extremum-seeking [GD15]. The effectiveness and efficiency of the optimiser depend on the properties of the optimisation problem at hand, particularly smoothness properties of the objective functions and nature of the design parameter set, that can be revealed by proper analysis of the problem.

2.6 Summary

Application of MPC in missile control is a potentially effective mean of improving engagement performance to address exigent mission requirements, primarily from increased target capabilities, that raise the need for an advanced control strategy which is both precise and robust. The notion is motivated by the innate abilities in handling non-linear dynamics, constraints and disturbance that are ingrained in the operation principles of MPC. Although a number of simulative studies have looked at applying MPC for missile control and demonstrated its advantages, many of the studies did not comprehensively consider the full autopilot and guidance control.

Although potentially performance-optimal, MPC-based controllers are sparsely used in practice due to the high computational power requirement for solving an optimisation problem at each sampling instant. This requirement is therefore an intrinsic design factor, directly associated with implementation cost of a capable processing hardware that is often not known a priori. Although an important element in MPC design, the implementation cost is traditionally often regarded as a constraint that is enforced when designing for an optimal performance. This approach is susceptible to sub-optimal overall design, for instance, when the controller is designed with over-specified hardware and over-meets mission requirements for performance, or under-designed such that the performance specifications are infeasible under the imposed implementation cost constraint. Setting best the cost constraint is thus a difficult task for the practitioner who aims to avoid unnecessary bills costed from over-design and/or the need for multiple design cycles in meeting mission requirements.

Multi-objective optimisation to maximise performance, as well as minimise implementation cost that often underlies application on fast systems, is potentially an effective design approach for MPC. The solution provides the practitioner a set of (Pareto) optimal choices to assist in the controller design process to balance performance and cost. This would avoid the need to pay unnecessary costs due to over-design or required re-design of the system that the current methods are susceptible to. There is novelty in the study of such a design technique for MPC, examining the tuning of structural control parameters for the optimisation of both performance and cost, as the current literature focusses

Controller	Parameters	Objectives		
		Performance		Performance & required cost
		Single-obj.	Multi-obj.	
PID	Gains	[ZN42]		
Sliding mode	Gains, manifold	[RP01]	[MTB14]	
MPC	Non-structural	[GS10]	[BKS14]	
	Structural	[BB14][a]		[Ker14][b]

[a]The study investigates the role of sampling time in linear quadratic regulator (not MPC).
[b]The study is a technical note for the co-design of real-time optimisation and does not provide comprehensive results for MPC.

Table 2.1: Overview in literature in controller calibration. A representative study is cited where applicable, not including the full list that has been discussed.

mostly on performance tuning and non-structural parameters. This is demonstrated in Table 2.1, where a brief overview of some of the studies in the literature of controller calibration is given, showing that the multi-objective optimisation of performance and required computational cost is still underdeveloped.

Finally, although general optimisers for a multi-objective optimisation approach are available, they lack in a targeted tack for controller design applications. The dependence of the MPC design objectives on the structural OCP parameters is non-trivial. Furthermore, the evaluation of closed-loop performance can be costly in time. An effective and efficient numerical optimiser is therefore required so that solutions can be obtained accurately and rapidly.

From the discussion above, the research gap in the current literature in the topics of this thesis is threefold,

1. A multi-objective design approach for MPC that considers closed-loop performance and implementation cost as dictated by required computational capacity is due to be fully investigated. Challenges arise from the fact that the two generally competing objectives are governed by a specific yet potentially complex trade-off relationship, and are dependent on multiple design parameters of different natures. Although the optimal co-design of engineering systems is recognised, the approach is yet to be comprehensively studied for MPC.
2. A specialised optimiser that targets the key properties of the formulated multi-objective MPC design optimisation problem is needed to effectively and effi-

ciently find the optimal trade-off set. Most well-known optimiser in the current literature are multi-purpose and generalised, potentially slow and inaccurate when used to solve the optimisation problem in hand. For a fast and accurate optimiser, the key properties of the problem – such as the relationship between the design objectives and parameters – need to be understood.

3. Recent developments in the integration of autopilot and guidance for high-performing missile control warrant for a controller architecture that can handle complex dynamics. The ability of MPC in explicitly handling the non-linearities, constraints and disturbance considered by the autopilot subsystem makes its role crucial in this area of research, but yet to be thoroughly demonstrated. Furthermore, guidance based on a receding horizon control that is needed for the integration with the autopilot is underdeveloped. Therefore, the full potential of MPC application in missile control has yet to be explored.

2.7 Research aims

To develop a design method for model-predictive control that optimises both closed-loop performance and required computational capacity

To formulate a multi-objective design approach for model-predictive control

The research aims to develop an approach that looks to directly address the multi-domain model-predictive control system design process without decomposition into subsystems. The scheme will be introduced as a multi-objective design approach for MPC (MOD-MPC) and formalised as a mathematical problem. The solution to the problem describes the optimal trade-off of control performance and required computational capacity of MPC by tuning of pertinent parameters.

To analyse of the MOD-MPC approach and develop an effective and efficient optimiser

After the mathematical formulation of the proposed MOD-MPC, the approach will be studied analytically. This is to reveal fundamental properties that define the nature of the solution to the approach as a mathematical problem. The presented properties form the basis for the development of an effective and efficient routine that can solve the proposed problem accurately and fast, providing practitioners of MOD-MPC a specialised tool for the application of the approach.

Illustrative results will be presented to demonstrate the proposed design scheme. Tests will be conducted to show the effectiveness and efficiency of the developed optimiser for MOD-MPC as compared to standard routines available in the current literature.

To demonstrate the multi-objective design approach on a proposed advanced missile control system

Finally, the MOD-MPC approach will be demonstrated for the design of an advanced missile control system. The control system will be based on a proposed model-predictive missile autopilot and guidance that builds on recent developments in advanced missile control.

Chapter 3
Multi-objective design of model-predictive control

A substantial proportion of this chapter has already appeared as an article in the journal Control Engineering Practice [BMMK16a].

IN this chapter, the multi-objective design of model-predictive control (MPC) approach is introduced and mathematically formalised. First, MPC is formulated in a closed-loop system that is to be optimised for performance. The associated optimal control problem is then introduced, which is the optimisation problem that needs to be solved in each sampling instant to control the plant of interest. After defining the control law, the design objectives of performance and the required computational capacity to implement the controller, as a measure of implementation cost, are presented. Control performance is represented by value functions that measure how well the system is controlled, for which a metric based on closed-loop simulations of the system is proposed. For the second objective, required computational capacity is defined by a non-dimensional number that represents computing power in multiples of that associated with the simulation hardware as the specifications benchmark. The design parameters for the system are described and discussed, onto which tuning is to be performed for an optimal balance of the design objectives. These parameters are classified as either coupled or decoupled given that they affect both or only one of the objective functions, respectively. The two design objectives are inherently competing, such that, under optimal conditions, a change in at least one of the pertinent

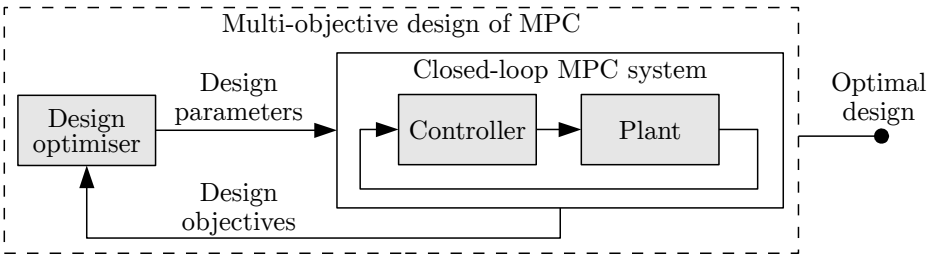

Figure 3.1: MOD-MPC schematic.

parameters that improves one objective comes at a comprise of the other. Once the elements of the closed-loop system are introduced, the chapter concludes by defining the design scheme as a multi-objective optimisation problem, whose the solution is the Pareto front of the problem that can provide useful insights on the trade-off between the two design objectives.

An illustrative overview of the design process is shown in Fig. 3.1. The closed-loop MPC system is designed/tuned via an outer optimisation loop to find the optimal design set that optimally balances the objectives over the calibration the chosen design parameters. The solution method for the outer optimisation problem depends on the optimiser used, which will be discussed in the next chapter.

3.1 Closed-loop control system

The closed-loop system is described by a controller-plant feedback framework. This is illustrated by the Controller and Plant subsystems in Fig. 3.1. Consider a non-linear plant model that describes the dynamics of a system of interest,

$$\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x}, \mathsf{u})$$

with states $\mathsf{x}(t) \in \mathbb{R}^{n_\mathsf{x}}$ and inputs $\mathsf{u}(t) \in \mathbb{R}^{n_\mathsf{u}}$. The dynamics satisfy the standard smoothness property of Lipschitz continuity, so that its first derivative is bounded. Lipschitz continuity implies that the dynamics of the system are uniquely defined for a given state. Moreover, the function is assumed to be continuously changing, either with non-continuous gradient i.e. class C^1, or a continuous gradient i.e. class C^2, depending on its differentiability.

Assumption 3.1 (Continuous $\mathsf{f}(\cdot)$)**.** $(\mathsf{x}, \mathsf{u}) \mapsto \mathsf{f}(\mathsf{x}, \mathsf{u})$ *is continuous in* (x, u) *and globally Lipschitz continuous in* x *uniformly in* u*.*

Assumption 3.2 (Differentiable $\mathsf{f}(\cdot)$)**.** $(\mathsf{x}, \mathsf{u}) \mapsto \mathsf{f}(\mathsf{x}, \mathsf{u})$ *is differentiable with respect to* u *for all* $\mathsf{x} \in \mathbb{R}^{n_\mathsf{x}}$*.*

Model-predictive control is implemented digitally, for which the system is discretised used such that the plant is controlled in a sampled-data fashion at sampling instants $t_i := ih$ for $i \in \mathbb{N}_{\geq 0}$ with sampling time h. With discretisation in mind, the control command sequence is restricted to a zero-order-hold (ZOH),

$$\mathsf{u}(t) = \mathsf{u}_i \qquad \forall t \in [ih, ih + h),\ i \in \mathbb{N}_{\geq 0}.$$

The aim is to control the plant by applying a control law κ to regulate the model to the origin. The control law depends on the current state $\mathsf{x}_i := \mathsf{x}(t_i)$ and the control design parameters $\mathbf{p}$,

$$\mathsf{u}_i = \kappa(\mathsf{x}_i, \mathbf{p}). \tag{3.1}$$

Let $\mathbf{p} := (p_1, \ldots, p_{n_p})$ contain the design parameters $p_1, \ldots, p_{n_p}$ to be tuned. Let $\mathbf{q} := (q_1, \ldots, q_{n_q})$ be the set of parameters that are fixed. To keep notational succinctness, dependence on the fixed parameter set $\mathbf{q}$ will not be explicitly stated as an argument throughout this thesis. The closed-loop system is thus defined by $\dot{\mathsf{x}}(t) = \mathsf{f}(\mathsf{x}, \kappa(\mathsf{x}_i, \mathbf{p})), \forall t \in [ih, ih + h), i \in \mathbb{N}_{\geq 0}$, or in discrete-time form,

$$\mathsf{x}_{i+1} = \mathsf{x}_i + \int_{t_i}^{t_i+h} \mathsf{f}(\mathsf{x}_i, \kappa(\mathsf{x}_i, \mathbf{p}))\, d\tau \qquad \forall i \in \mathbb{N}_{\geq 0} \tag{3.2}$$

where $\mathsf{x}_i = \mathsf{x}(ih)$.

Tracking formulation

Many real-world applications, including those that are studied in this thesis, are more appropriately formulated as a tracking problem, where the controller has to drive the state $\widetilde{\mathsf{x}}$ to some steady state $\widetilde{\mathsf{x}}_\mathrm{o}$ with some plant model

$$\dot{\widetilde{\mathsf{x}}} = \widetilde{\mathsf{f}}(\widetilde{\mathsf{x}}, \mathsf{u}).$$

To transform the tracking problem into that of regulation, a simple coordinate transformation can be performed,

$$\mathsf{x} := \widetilde{\mathsf{x}} - \widetilde{\mathsf{x}}_\mathrm{o} \qquad \widetilde{\mathsf{f}}(\mathsf{x} + \widetilde{\mathsf{x}}_\mathrm{o}, \mathsf{u}) = \mathsf{f}(\mathsf{x}, \mathsf{u}) \tag{3.3}$$

to define an error system that is to be regulated to zero.

3.2 Optimal control problem – general (non-linear) MPC

In MPC, the control command is obtained by solving a finite-horizon, optimal control problem (OCP) at each sampling instant t_i given the current state of the plant x_i. A common form of the OCP that will also be used throughout the study is given in the following

$$(x^*(\cdot), u^*(\cdot)) := \arg\min_{(x,u)} J(x, u, \mathbf{p}) \tag{3.4a}$$

$$\text{s.t. } x(0) = \mathsf{x}_i \tag{3.4b}$$

$$\dot{x}(\tau) = f(x(\tau), u(\tau), \mathbf{p}) \qquad \forall \tau \in [0, T] \tag{3.4c}$$

$$x(\tau) \in \mathcal{X} := [\underline{x}, \overline{x}]\,,\ u(\tau) \in \mathcal{U} := [\underline{u}, \overline{u}] \qquad \forall \tau \in [0, T) \tag{3.4d}$$

$$u(\tau) = u(kh) =: u_k,\ \forall k \in \mathbb{N}_{\geq 0} \qquad \forall \tau \in [kh, kh + h). \tag{3.4e}$$

For succinctness, the dependence of the optimal solution $x^*(\cdot)$ and $u^*(\cdot)$ on $(\mathsf{x}_i, \mathbf{p})$ is omitted here onwards unless needed for clarity. The real-time variable x is distinct from the predicted variable x used internally in the OCP, although sized equally such that $x(t) \in \mathbb{R}^{n_\mathsf{x}}$ and inputs $u(t) \in \mathbb{R}^{n_\mathsf{u}}$. The true plant model $\mathsf{f}(\cdot)$ is also distinct to the prediction model $f(\cdot)$ used internally in the OCP. The two models have the same equilibrium at the origin, that is $f(0, 0) = \mathsf{f}(0, 0)$. The optimisation is subject to the prediction model (3.4c) representing the dynamics of the plant initialised at (3.4b), as well as the plant constraints (3.4d) that are defined as rectangular sets, with upper- and lower-bounds on the predictive states and inputs. The zero-order-hold control (3.4e) discretises the control command over the sampling steps $k \in \{0, \ldots, N-1\}$. This is the general control problem in which the constraints are non-linear, in the form of non-linear prediction model (3.4c) and/or non-linear state/input constraints (3.4d). For this reason, the controller associated with such and OCP will be labelled non-linear MPC (NMPC).

The OCP solution represents the optimal evolution of the plant within the prediction horizon for the given cost function. The first part of the control command will be applied in each sampling time such that the control law is defined by

$$\kappa(\mathsf{x}_i, \mathbf{p}) := u^*(0). \tag{3.5}$$

In most cases, the cost function is made up of the stage cost J_s and terminal cost J_t,

$$J(x, u, \mathbf{p}) := \int_0^{T(\mathbf{p})} J_\mathsf{s}(x(\tau)), u(\tau), \mathbf{p})\ d\tau + J_\mathsf{t}(x(T), u(T), \mathbf{p}), \tag{3.6}$$

where $T(\mathbf{p})$ is the prediction horizon length dictating how far prediction is made ahead on the dynamics of the controlled plant. In particular, a quadratic form

$$J_\mathsf{s}(x, u, \mathbf{p}) := \|x(\tau)\|_Q^2 + \|u(\tau)\|_R^2, \qquad J_\mathsf{t} := \|x(T)\|_{Q_\mathsf{t}}^2 \tag{3.7}$$

is predominantly used to penalise the state/input deviations from zero.

Remark 3.1. The stage cost is weighted by a positive semi-definite Q and positive definite R, and the terminal cost is weighted by a positive semi-definite Q_t, so that the cost function is positive definite.

Discrete-time form

Since the input is constrained with ZOH, the OCP can be recast in discrete-time form. The prediction horizon T is divided into N prediction steps, each with an interval length of the sampling time h. The predicted states and input are presented in discrete-time with the vectors

$$\begin{aligned} \mathbf{x} &:= (x_0, \dots, x_N) & \mathbf{u} &:= (u_0, \dots, u_{N-1}) \\ \text{where} \quad x_k &:= x(kh) & u_k &:= u(kh) \end{aligned} \tag{3.8}$$

as per (3.4e). This yields the discrete-time OCP (dOCP),

$$(\mathbf{x}^*, \mathbf{u}^*) := \arg\min_{(\mathbf{x},\mathbf{u})} \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{3.9a}$$

$$\text{s.t.} \quad x_0 = \mathsf{x}_i \tag{3.9b}$$

$$x_{k+1} = \overline{f}(x_k, u_k, \mathbf{p}) := x_k + \int_{t_i}^{t_i+h} f(x(\tau), u_k, \mathbf{p})\, d\tau \quad \forall k \in \{0, \dots, N-1\} \tag{3.9c}$$

$$x_k \in [\underline{x}, \overline{x}]\,, \; u_k \in [\underline{u}, \overline{u}] \qquad \forall k \in \{0, \dots, N-1\}. \tag{3.9d}$$

In this form, the cost function (3.9a) is cast in a discrete-time form,

$$\overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) := \sum_{k=0}^{N-1} \left\| \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}} + \|x_N\|^2_{Q_t}. \tag{3.10}$$

The control law is given by

$$\kappa(\mathsf{x}_i, \mathbf{p}) := u_0^*. \tag{3.11}$$

Remark 3.2. It is possible for the discrete-time cost function to be made equivalent to the continuous-time counterpart,

$$\overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = J(x, u, \mathbf{p})$$

where the states and input follow (3.8). For this to be true, the cost weights must satisfy specific relationships as outlined later in (3.18) in the case of a linear prediction model. When the cost functions are equivalent, equivalence in the con-

trol law

$$u_0^* = u^*(0)$$

as per (3.5) and (3.11) is also achieved.

While the continuous-time OCP require more general optimisation routines to numerically solve, discretisation of the OCP allows for the use of more specific numerical algorithms that is often lighter in computational load, allowing for faster solution times. The non-linear dOCP can be solved using a sequential quadratic programming (SQP) algorithm, the details of which are given in Appendix B.1.

3.3 The OCP as a quadratic program – MPC

The OCP (3.4) is defined generally as a non-linear program. It is common for the problem to be reduced to a quadratic program (QP). The cost function used is quadratic, and state and input constraints (3.4d) are linear, leaving only the prediction model (3.4b) that must be linearised for the problem to reduce to a QP. Specifically, the OCP in a reduced QP form (Q-OCP) is defined as

$$(x^*(\cdot), u^*(\cdot)) := \arg\min_{(x,u)} J(x, u, \mathbf{p}) \tag{3.12a}$$

$$\begin{aligned} \text{s.t. } & x(0) = \mathsf{x}_i \\ & \dot{x}(\tau) = Ax(\tau) + Bu(\tau) && \forall \tau \in [0, T] \\ & x(\tau) \in [\underline{x}, \overline{x}], u(\tau) \in [\underline{u}, \overline{u}] && \forall \tau \in [0, T) \end{aligned} \tag{3.12b}$$

where a linear time-invariant prediction model (3.12b) is used, following

$$A := \left.\frac{\partial \mathsf{f}}{\partial \mathsf{x}}\right|_{0,0} \qquad\qquad B := \left.\frac{\partial \mathsf{f}}{\partial \mathsf{u}}\right|_{0,0}.$$

Consequently, the associated controller will be labelled linear MPC, or simply MPC. When the transformation of tracking into regulation (3.3) is applicable, the constraint bounds will also shifted according to

$$\underline{x} = \underline{\widetilde{\mathsf{x}}} - \widetilde{\mathsf{x}}_\mathrm{o} \qquad\qquad \overline{x} = \overline{\widetilde{\mathsf{x}}} - \widetilde{\mathsf{x}}_\mathrm{o}. \tag{3.13}$$

The cost function (3.12a) remains the quadratic that is used in the general non-linear OCP $J(x, u, \mathbf{p}) := \int_0^{T(\mathbf{p})} \|x(\tau)\|_Q^2 + \|u(\tau)\|_R^2 \; d\tau + \|x(T)\|_{Q_\mathrm{t}}^2$ with its constituent stage and terminal cost terms.

Discrete-time form

As with the case of the general OCP, the Q-OCP can be recast in discrete-time form,

$$(\mathbf{x}^*, \mathbf{u}^*) \coloneqq \arg\min_{(\mathbf{x},\mathbf{u})} \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{3.14a}$$

$$\text{s.t. } x_0 = \mathsf{x}_i \tag{3.14b}$$

$$x_{k+1} = \overline{A}x_k + \overline{B}u_k \qquad \forall k \in \{0, \dots, N-1\} \tag{3.14c}$$

$$x_k \in [\underline{x}, \overline{x}], u_k \in [\underline{u}, \overline{u}] \qquad \forall k \in \{0, \dots, N-1\}. \tag{3.14d}$$

Note that the dependence of the optimal solution $\mathbf{x}^*$ and $\mathbf{u}^*$ on $(\mathsf{x}_i, \mathbf{p})$ is omitted.

The discretisation of the OCP can be generalised for the case when the prediction horizon length is not an integer multiple of the sampling time, $T \neq Nh$. In this case, a residual sampling time

$$h_\mathrm{r} \coloneqq T - Nh$$

is used, where the number of prediction steps is determined by

$$N = \lfloor T/h \rfloor. \tag{3.15}$$

The residual sampling step is required so that the discretised OCP is equivalent to its continuous-time counterpart. The discrete OCP with the residual sampling step is then given by the dQ-OCP

$$(\mathbf{x}^*, \mathbf{u}^*) \coloneqq \arg\min_{(\mathbf{x},\mathbf{u})} \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{3.16a}$$

$$\text{s.t. } x_0 = \mathsf{x}_i \tag{3.16b}$$

$$\left.\begin{aligned} x_{k+1} &= \overline{A}x_k + \overline{B}u_k \qquad \forall k \in \{0, \dots, N-1\} \\ x_\mathrm{t} &= \overline{A}_\mathrm{r}x_N + \overline{B}_\mathrm{r}u_N \end{aligned}\right\} \tag{3.16c}$$

$$x_k \in [\underline{x}, \overline{x}], u_k \in [\underline{u}, \overline{u}] \qquad \forall k \in \{0, \dots, N\}. \tag{3.16d}$$

The cost function is given as

$$\overline{J}(x, u, \mathbf{p}) \coloneqq \sum_{k=0}^{N-1} \left\| \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}} + \overline{J}_\mathrm{r} + \|x_\mathrm{t}\|^2_{\overline{Q}_\mathrm{t}}$$

where residual cost term is either valued as

1. $\overline{J}_{\mathrm{r}} = 0$ with $x_{\mathrm{t}} = x_N$ for (3.14), or
2. $\overline{J}_{\mathrm{r}} = \left\| \begin{bmatrix} x_N \\ u_N \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q}_{\mathrm{r}} & \overline{S}_{\mathrm{r}} \\ \overline{S}_{\mathrm{r}}^{\mathsf{T}} & \overline{R}_{\mathrm{r}} \end{bmatrix}}$ for (3.16).

The generalisation of the prediction horizon being a non-integer multiple of the sampling time can be made for the non-linear d-OCP (3.9), but is left unwritten due to its triviality.

Following the equivalence between the OCP and dOCP, the Q-OCP can be made equivalent to dQ-OCP so that the solutions of the two problems are identical. For this to hold, the discrete-time transition matrices must be equivalent to the linear dynamics as follows,

$$\overline{A} := \Phi(h) := e^{Ah} \qquad \overline{B} := \Gamma(h) := \int_0^h e^{A(h-\tau)}\, d\tau\, B \tag{3.17}$$

with the residual transition matrices

$$\overline{A}_{\mathrm{r}} := \Phi(h_{\mathrm{r}}) \qquad \overline{B}_{\mathrm{r}} := \Gamma(h_{\mathrm{r}}).$$

For the discrete-time cost function to be equivalent to the continuous-time counterpart (see Remark 3.2), the weights are set to satisfy the following relationships,

$$\overline{Q} := \int_0^h \|\Phi(\tau)\|^2_Q\, d\tau \quad \overline{R} := \int_0^h \|\Gamma(\tau)\|^2_Q\, d\tau + hR \quad \overline{S} := \int_0^h \Phi(\tau)^{\mathsf{T}} Q \Gamma(\tau)\, d\tau \tag{3.18}$$

such that the solution obtained from (3.14) is identical to that of (3.12). The residual cost weighting terms are given by

$$\overline{Q}_{\mathrm{r}} := \int_0^{h_{\mathrm{r}}} \|\Phi(\tau)\|^2_Q\, d\tau \quad \overline{R}_{\mathrm{r}} := \int_0^{h_{\mathrm{r}}} \|\Gamma(\tau)\|^2_Q\, d\tau + hR \quad \overline{S}_{\mathrm{r}} := \int_0^{h_{\mathrm{r}}} \Phi(\tau)^{\mathsf{T}} Q \Gamma(\tau)\, d\tau.$$

In QP form, the dQ-OCP can be readily solved with relevant algorithms, for example interior point methods [BV04]. Table 3.1 summarises the different forms

Name	Label	Timeframe	Problem type
OCP	(3.4)	Continuous	Non-linear
dOCP	(3.9)	Discrete	Non-linear
Q-OCP	(3.12)	Continuous	Quadratic
dQ-OCP	(3.14) or (3.16)	Discrete	Quadratic

Table 3.1: OCP forms.

of OCP that have been outlined. Here onwards, the name OCP will be interchangeably used for all four forms, unless a specific form needs to be referred to for clarity.

3.4 Design objectives

Having now formulated the closed-loop system, the objectives of the controller design can be formalised. The study will focus on two metrics that are underlying to model-predictive controllers. The first objective is control performance, measuring the ability of the controller to regulate/track the plant. The second objective is the required computational capacity which is pivotal especially in the case of the computationally heavy MPC. This metric serves as an indicative measure of the overall cost to implement the controller.

Intuitively, the two design objectives of performance and cost are often competing, in the sense that improving one objective requires compromise of the other. The optimal trade-off relationship between the two objectives is known as *Pareto* optimality. This chapter will focus on introducing and formalising the design objectives. Analysis of the competing relationship between the two objectives will be explored in the next chapter.

3.4.1 Control performance

For the established model-predictive controller (3.4) which aims to regulate the plant model, performance is measured by how well the controller steers the states to the origin over time. An indicative measure can be obtained from the *open-loop* value function, either in the non-linear OCP case,

$$\begin{aligned} J^*(\mathsf{x}_i, \mathbf{p}) &:= J(x^*(\mathsf{x}_i, \mathbf{p}), u^*(\mathsf{x}_i, \mathbf{p}), \mathbf{p}) \\ &= \overline{J}(x^*(\mathsf{x}_i, \mathbf{p}), u^*(\mathsf{x}_i, \mathbf{p}), \mathbf{p}) \\ &\text{where } (x^*, u^*) \text{ is obtained from (3.4) or (3.9)} \end{aligned}$$

or the quadratic case,

$$\begin{aligned} \overline{J}^*(\mathsf{x}_i, \mathbf{p}) &:= J(x^*(\mathsf{x}_i, \mathbf{p}), u^*(\mathsf{x}_i, \mathbf{p}), \mathbf{p}) \\ &= \overline{J}(x^*(\mathsf{x}_i, \mathbf{p}), u^*(\mathsf{x}_i, \mathbf{p}), \mathbf{p}) \\ &\text{where } (x^*, u^*) \text{ is obtained from (3.12), (3.14), or (3.16).} \end{aligned} \tag{3.19}$$

These are the values of the OCP cost function at the optimal solution.

In most applications, control is implemented in *closed-loop* instead of open-loop so that control actions are calculated based on the feedback from the plant. Although open-loop applications exist, most systems requires closed-loop control. The open-loop value function is a good indicative measure for performance and has been used in previous studies [BKS14,BB14]. A measure based on closed-loop simulation, however, might be a more accurate representation of performance. Subsequently, closed-loop performance metrics can be obtained through a simulation of the MPC system to draw various numerical measures. The most common of these is the integrated squared error (ISE) of the states and inputs to measure regulating performance.

The closed-loop performance is associated with a given initial condition

$$\mathsf{x}_0 \qquad \text{or error state} \qquad \mathsf{x}_0 = \widetilde{\mathsf{x}}_0 - \widetilde{\mathsf{x}}_\mathrm{o} \tag{3.20}$$

and defined as a value-function based on ISE as follows

$$U^*(\mathsf{x}_0, \mathbf{p}) \coloneqq \int_0^\infty U(\mathsf{x}(\tau), \mathsf{u}(\tau))\, d\tau \tag{3.21a}$$

$$\text{s.t. } \mathsf{x}(0) = \mathsf{x}_0 \tag{3.21b}$$

$$\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x}, \mathsf{u}) \tag{3.21c}$$

$$\mathsf{u}(\tau) = \kappa(\mathsf{x}(t_i), \mathbf{p}) = u_0^* \qquad \forall \tau \in [ih, ih + h),\ \forall i \in \mathbb{N}_{\geq 0}. \tag{3.21d}$$

The cost function is chosen as a quadratic

$$U(\mathsf{x}, \mathsf{u}) \coloneqq \|\mathsf{x}(\tau)\|_Q^2 + \|\mathsf{u}(\tau)\|_R^2 \tag{3.22}$$

in accordance with the stage cost of the MPC cost function (3.6) with equivalent weights (3.7). The closed-loop value function provides a numerical measure of the control performance of the MPC law with the associated OCP.

Assumption 3.3 (Recursive feasibility)**.** *The closed-loop system* (3.21) *is recursively feasible such that for a given initial state* $\mathsf{x}_0 \in \mathcal{X}_\mathrm{S}$*, the control law* (3.21d) *is feasible and remains feasible at all subsequent sampling steps.* $\mathcal{X}_\mathrm{S}$ *is the associated feasible set.*

Assumption 3.4 (Stability)**.** *The control law* (3.21d) *guarantees the stability of the closed-loop system* (3.21)*.*

Stability of the closed-loop system can be guaranteed by an appropriate choice of the OCP parameters. Some examples are given in the following.

1. It has been noted in [LSA71] that there exist *critical sampling times* where the discrete-time plant loses full controllability, which consequently leads to an unstable closed-loop trajectory. Since fast systems are of interest, the sampling times considered can safely be assumed to not include critical sampling times.
2. For $f(\cdot) = \mathsf{f}(\cdot)$, an appropriate choice of Q_t in (3.7) is able to guarantee stability [May14]. The restriction on model fidelity might be relaxed so that the discrepancy between the two is bounded. That is, for a given $x(0) = \mathsf{x}(0)$ and u, if $x(t)$ is a solution of $\dot{x} = f(x, \mathsf{u})$ and $\mathsf{x}(t)$ is a solution of $\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x}, \mathsf{u})$ in $t \in [0, T]$, then $\|x(t) - \mathsf{x}(t)\| < \varepsilon$, $\forall t \in [0, T]$ for some bound ε.

Exact conditions that guarantee feasibility and stability as per Assumptions 3.3 and 3.4 depend on the different choices of the control law and assumptions on the plant model. Specification of these conditions is amply available in the literature, for instance [KM00, MR09, PZ95].

A combination of the closed-loop value function for a range of initial conditions $\mathcal{X}_0$ can be used as a more comprehensive measure of control performance. For such a metric, $\mathcal{X}_0$ is the set of initial conditions representative of the intended operating range of the controlled plant, each of which is weighed by w_i to determine the relative significance of each scenario. A linear combination can be used as given in the following

$$V^*(\mathcal{X}_0, \mathbf{p}) \coloneqq \sum_{\mathsf{x}_0 \in \mathcal{X}_0} w_i U^*(\mathsf{x}_0, \mathbf{p}). \tag{3.23}$$

A linear combination is chosen to preserve the smoothness properties of the value-function U^*.

3.4.2 Required computational capacity

In designing the controller, the capability of the computational hardware onto which the controller will be implemented is often not known a priori. It is therefore imperative to treat the required computational capacity/resource, which is a fundamental factor in determining cost, as a design objective.

In this study, numerical simulations are conducted instead of directly testing the control plant to evaluate the closed-loop performance of the controller. The computational data obtained in simulation is reflective of the hardware used in the simulation platform, namely the *simulation hardware*. This is to be differ-

entiated from the *implementation hardware*, which is the actual platform used to implement the controller.

A number of pertinent parameters dictate the computational complexity of the OCP (3.4). In turn, the complexity governs the required capacity, affecting the time taken by a processing unit to generate a control command. For real-time implementation, the *solution time*, i.e. the time taken for a command input to be generated numerically, is upper-bounded by the sampling time h. Let the upper-bound for the solution time for the simulation hardware and a given set of design parameters be $\zeta(\mathbf{p})$. This indicates the computational complexity for a given design $\mathbf{p}$. Based on this, a dimensionless measure denoted the *utilisation (capacity) number* can be derived,

$$\eta(\mathbf{p}) \coloneqq \zeta(\mathbf{p})/h.$$

The utilisation number indicates the required computational power of the implementation hardware relative to that used in the numerical simulation. Specifically, η is a multiplier of the simulation hardware capacity that defines the minimum capability needed so that the controller can be functionally implemented.

For the utilisation number to be a meaningful measure, it is necessary to define the relationship between the simulation and implementation hardware. Consequently, the respective solution time upper-bound is assumed to be linearly proportional between the simulation and implementation hardware. The importance of this assumption will be clarified later in Proposition 3.1 when the multi-objective design problem is formulated.

Assumption 3.5 (Scalability of solution time upper-bound)**.** *The solution time upper-bound of the simulation hardware ζ is scalable to the solution time upper-bound ζ_{I} if the implementation hardware was used. That is, $\zeta_{\mathrm{I}} = a\zeta$ for some constant multiplier $a \in \mathbb{R}_{>0}$.*

A measure ζ_{FLOP} specifying the number of floating-point operations to solve the OCP can be used, instead of the temporal measure ζ, to indicate the required computational capacity to practically implement the controller. In this case, the required computational capacity can be indexed by $\eta_{\mathrm{FLOP/s}} = \xi/h$. Here, $\eta = 1$ is equivalent to $\eta_{\mathrm{FLOP/s}}$ being equal to the floating-point operations per second (FLOP/s) capability of the simulation hardware.

3.5 Design parameters

A number of design parameters can be identified in the formulated model-predictive controller. A key design parameter is the sampling time h that dictates how often a new control input can be commanded to the plant. The interval also sets the upper-bound on the time available for the computing hardware to solve the OCP. Next, the length of the prediction horizon T is the time period in which constraints can be applied in the prediction of the future plant behaviour, and thus its value affects the performance of the controller. In a discrete-time system, the horizon is typically defined by the number of prediction steps N such that $T = Nh$. Other than governing the prediction horizon length, the number of prediction steps directly affects the size of the discrete-time OCP as it dictates the number of unknowns in the problem. Another design parameter is the model type which is the choice of the prediction model $f(\cdot)$, which is already shown to be able to take a linear or non-linear form.

Attributes of the cost function are also design parameters. For instance, in most forms of cost function, these are cost weighting matrices that needs to be set. These affect the solution of the OCP at a given sampling instant and hence the control performance. Variations in cost function attributes also affect the numerical conditioning of the OCP. Depending on the chosen solver algorithm, particularly its degree of sensitivity to ill-conditioning, the effect of cost function attributes on the time taken to numerically solve the OCP may be assumed to be negligible. On the other hand, the solver algorithm used to solve the OCP is a design choice that affects computational complexity more so than control performance. Control performance dependence on the solver arises from the fact that there might be multiple local minima associated with the non-linear (d)OCP such that a different solver might converge to a different minimum. The linear (d)QOCP, is a convex problem, and so as long as the solver is convergent, it can be assumed to find the one local (global) minimum of the OCP.

The choice of algorithm is closely related to the representation of the OCP, for example, a dense or a sparse representation. Some algorithms are suited to a dense representation whereas others handle a sparsely presented OCP better. Both algorithm and representation, however, can be assumed to not affect control performance so that the same control command is produced regardless of the al-

Design parameter	Symbol(s) (if applicable)	Effect on [a] V^*	Effect on [a] η	Association [b]
Cost function attributes	Q, R, and Q_t	✓	? [c]	Algorithm
Sampling time	h	✓	✓	
Prediction horizon length	T or N	✓	✓	
Prediction model type [d]	$f(\cdot)$	✓	✓	
State/input constraints type[d]	$\mathcal{X}$ and $\mathcal{U}$	✓	✓	
OCP representation			✓	
Solver algorithm		?	✓	Implementation
Solver tolerance		✓	✓	
Numerical precision		✓	✓	

[a] V denotes control performance, η denotes required computational capacity (cost).
[b] Parameters are classified based on association to the algorithm or implementation of the OCP.
[c] Question mark denotes that the effect depends on the choice of other parameters.
[d] Prediction model and state/input constraints type constitute constitute the OCP constraints.

Table 3.2: Design parameters of the OCP.

gorithm and/or OCP representation used. Finally, a numerical tolerance can be used as an algorithm attribute, dictating the accuracy and speed at which the numerical solution of the OCP is obtained. That is, suboptimal solutions can be used to accommodate fast control sampling rates [SMR99,ZJM11]. Thus, the parameter affects both the control performance and required computational capacity.

The aforementioned design parameters are outlined in Table 3.2 and classified in accordance to their effect on the design objectives and their association to either the control algorithm or implementation. In multi-objective design, focus will be given to design parameters that are *coupled*, which are those that affect both design objectives. On the other hand, *decoupled* parameters are those that only affect one design objective and have no effect on the other, as well as those that affect both objectives only if certain choices for the other parameters are picked. This study focuses on parameters that are associated with the algorithm, so that parameters that exclusively are implementation attributes, such as numerical precision [Ker14], are not considered even though they affect both objectives. Consequently, the proposed design problem will consider only parameters of the OCP that are coupled, denoted as *structural* design parameters. Decoupled parameters and/or those associated to numerics will be treated as fixed.

3.6 Choice of design parameters

The design parameters that are to be calibrated are coupled and attributed to the algorithm of the OCP. These parameters will be denoted as the *structural* design parameters, which include the sampling time, length of prediction horizon, as well as the type of prediction model and state/input constraints. Focus is given on the two underlying structural parameters of the OCP, namely the sampling time and prediction horizon length, and the prediction model type is treated as constant. This is first motivated by the fact that the two design parameters are closely correlated. As a consequence, the resulting system would have specific characteristics that can be explored and revealed, allowing for a targeted numerical optimiser to be proposed.

The sampling time and length of prediction horizon are the design parameters, the latter of which can be determined by the choice of number of prediction steps N with the definition that

$$T = Nh. \tag{3.24}$$

Consequently, the design parameters are set as

$$\begin{aligned} \mathbf{p} &= (h, N) \in \mathcal{P} \\ \text{where } \mathcal{P} &:= \mathbb{R}_{>0} \times \mathbb{N}_{>0}. \end{aligned} \tag{3.25}$$

In this study, the prediction model is defined to be linear-time invariant, a choice that is especially relevant in real-world applications as it reduces the general non-linear OCP into a quadratic program, a convex optimisation problem for which many practical solvers exist as numerical routines for QPs have been much developed [BV04]. Further, state and input constraints are prescribed to be upper- and lower- bounds, such that the associated sets are rectangular. Consequently, the analysis throughout the chapter refers to the dQ-OCP (3.14) (or, in the case where $T = Nh$ is not imposed, (3.16)). The associated controller is the common (linear-quadratic) MPC, which is widely used in practice due to the convexity of the problem allowing for necessary and sufficient conditions for optimality to be established rather straightforwardly. This in turn allows for smoothness properties to be founded as given in Section 4.2.2 in the next chapter.

Remark 3.3. The solution time, as defined in Section 3.4.2, depends on a number of design parameters of the OCP as a quadratic program, given the choice of a linear time-invariant prediction model. This dependence is outlined in Table 3.3. Given that the cost function is positive definite (Remark 3.1), QP optimisation is known to be a P problem so that is solvable in polynomial time [KTK80] as its size increases. The size of the OCP is directly associated with the number of unknowns, which is directly proportional to the number of prediction steps N. Consequently, the solution time of the OCP as a quadratic program will be modelled as a polynomial with respect to N.

Assumption 3.6 (Solution time upper-bound)**.** *The upper-bound on solution time ζ is monotonically increasing with the number of prediction steps N, modelled by a polynomial of degree n_a,*

$$\zeta(\mathbf{p}) \coloneqq \sum_{i=0}^{n_a} a_i(\mathbf{p}) N^i \tag{3.26}$$

for some constants $a_i(\mathbf{p})$, $i \in \{0, \ldots, n_a\}$ that depend on design parameters other than N.

The other parameters affect the required computational capacity in less obvious ways and are aggregated into the constants $a_i(\mathbf{p})$. Consequently,

$$\eta(\mathbf{p}) \coloneqq \frac{\zeta(\mathbf{p})}{h} \coloneqq \frac{1}{h} \sum_{i=0}^{n_a} a_i N^i. \tag{3.27}$$

Parameters associated with the implementation aspect of the system are fixed. Consequently, the dependence of the constants a_i in the model of the solution

Design parameter	Effect on solution time (3.26)
Number of prediction steps	Size of OCP
Cost function attributes[a]	Matrices of the OCP and their conditioning
Sampling time[a]	Matrices of the OCP and their conditioning
Prediction model type[a,b]	OCP form, either an NLP or QP
State/input constraints type[a,b]	OCP form, either an NLP or QP
OCP representation[b]	No. of constraints and unknowns of the OCP

[a] Aggregated in the constants a_i in (3.26).
[b] Prediction model and state/input constraints type constitute constitute the OCP constraints.

Table 3.3: Design parameters affecting solution time.

time upper-bound ζ (3.26) on these design parameters not part of the consideration. These constants are aggregations of the dependence of the solution time upper-bound on design parameters other than the number of prediction steps N, including design parameters that are attributed to the OCP which are the cost function attributes, sampling time, type of constraints and representation of the OCP as outlined in Table 3.3. The chosen solver algorithm is assumed to be insensitive to numerical conditioning of the OCP, so that sampling time and attributes of the cost function are assumed to not affect solution time. Since the rest of the analytical design parameters are fixed along with those associated with implementation, the constants a_i are assumed to be independent of $\mathbf{p}$.

Assumption 3.7. *The constants a_i in* (3.26) *are independent of the chosen design parameters in* $\mathbf{p}$.

Finally, given that the solver algorithm, tolerance and numerical precision are well-chosen, the solver is convergent and able to accurately find the solution of the chosen OCP. As discussed in the introduction of the solver algorithm as a design parameter, control performance can be assumed to be independent of the OCP solver in the case of dQ-OCP, as long as the solver is convergent so that it will find the one local (thus global) minimum of the OCP as a convex problem.

Assumption 3.8. *The numerical solution of the OCP obtained using the chosen solver algorithm, tolerance, and numerical precision is close to the true/analytical solution.*

3.7 Multi-objective design of MPC

Let the design objectives be contained in a vector

$$\boldsymbol{\ell}(\mathbf{p}) = (V^*(\mathbf{p}), \eta(\mathbf{p}))$$

for $\boldsymbol{\ell} \in \mathbb{R}^{n_\ell}$ and $n_\ell = 2$. In this case, the vector contains the closed-loop control performance[1] and required computational capacity. The multi-objective design of MPC (MOD-MPC) is posed as the following,

$$\mathcal{P}_\bullet(\mathcal{P}_\mathrm{s}) \coloneqq \arg\operatorname*{m-min}_{\mathbf{p}} \boldsymbol{\ell}(\mathbf{p}) \tag{3.28a}$$

$$\text{s.t. } \mathbf{p} \in \mathcal{P}_\mathrm{s} \subset \mathcal{P}. \tag{3.28b}$$

[1] $J^*(\mathbf{p})$ is used instead of $V^*(\mathbf{p})$ for the first objective in the case of open-loop evaluations.

The minimisation (denoted m-min) is a multi-objective optimisation to find the Pareto optimal design set $\mathcal{P}_\bullet$ for a given search space $\mathcal{P}_\mathrm{s}$. The search space is a bounded set, that is a subset of the open set $\mathcal{P}$, defining the range of the parameter to be explored by the optimiser[2] for the problem. This solution set contains the Pareto optimal design choices for the practitioner to select from, based on the Pareto front

$$\mathcal{L}(\mathcal{P}_\mathrm{s}) \coloneqq \left\{\boldsymbol{\ell}(\mathbf{p}),\, \forall \mathbf{p} \in \mathcal{P}_\bullet(\mathcal{P}_\mathrm{s})\right\} \tag{3.29}$$

that shows the optimal trade-off between the two objectives. This is contrasted to that of a single-objective optimisation where the solution correspond to one minimum objective point. The Pareto optimal points that make up the Pareto front are said to be *non-dominated*, as per Definitions 1.4 and 1.5.

Having defined Pareto optimality of an evaluation point, the importance of Assumption 3.5 that defines the scalability between the simulation and implementation hardware is now clearer.

Proposition 3.1. *Given satisfaction of Assumption 3.5, then a Pareto optimal point obtained using the simulation hardware stays Pareto optimal for the implementation hardware.*

Proof. Let the utilisation number η associated with the solution time upper-bound model ζ for the simulation hardware be given as $\eta \coloneqq \zeta/h$ as per (3.27). Suppose that the utilisation number associated with the implementation hardware is denoted by $\eta_\mathrm{I} \coloneqq \zeta_\mathrm{I}/h$. Given that Assumption 3.5 holds, $\eta_\mathrm{I} = a\zeta/h = a\eta$ for some positive constant $a \in \mathbb{R}_{>0}$. Now consider a point $\boldsymbol{\ell}_\bullet = (V^*_\bullet, \eta_\bullet)$ that is Pareto optimal within some set of points L. The point $\boldsymbol{\ell}_{\mathrm{I}\bullet} = (V^*_\bullet, a\eta_\bullet) = (V^*_\bullet, \eta_{\mathrm{I}\bullet})$ and set $L_\mathrm{I} = \{\boldsymbol{\ell}_\mathrm{I} \coloneqq (V^*, \eta_\mathrm{I}) \,|\, \eta_\mathrm{I} = a\eta,\, \forall (V^*, \eta) \in L\}$ denotes the equivalent in the implementation environment. For any point $\boldsymbol{\ell}_\mathrm{I} \in L_\mathrm{I}$ to dominate $\boldsymbol{\ell}_{\mathrm{I}\bullet}$, the inequality $\eta_\mathrm{I} \leq \eta_{\mathrm{I}\bullet}$ must hold (see Definition 1.4). That is, $a\eta \leq a\eta_\bullet$ or $\eta \leq \eta_\bullet$, which cannot be true since $\eta_\bullet$ is associated with the Pareto optimal point $\boldsymbol{\ell}_\bullet$ in L. By this contradiction, it is implied that there is no point in L_I that dominates $\boldsymbol{\ell}_{\mathrm{I}\bullet}$, and so $\boldsymbol{\ell}_{\mathrm{I}\bullet}$ is Pareto optimal within L_I. This is illustrated in Fig. 3.2. □

[2]To establish clarity and consistency, the term *optimiser* is associated with the multi-objective design optimisation whilst *solver* is used for the OCP optimisation throughout the thesis.

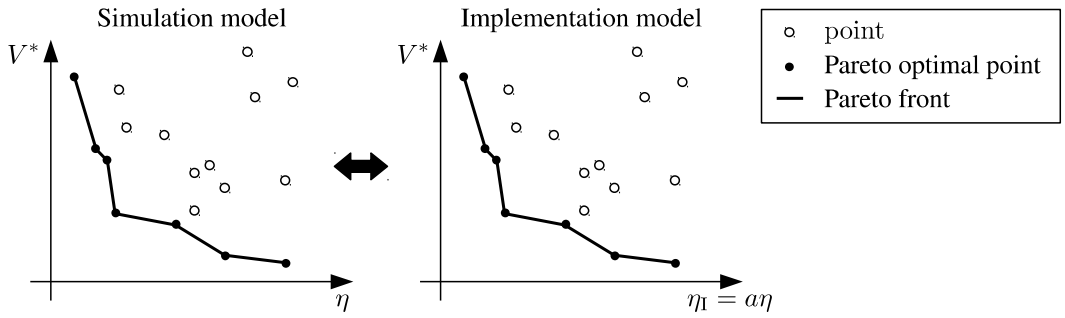

Figure 3.2: Illustration of the implication of the scalability of the solution time upper-bound between simulation and implementation, $\zeta_{\mathrm{I}} = a\zeta$ as per Assumption 3.5. Pareto optimality is retained between the two cases.

The Pareto front $\mathcal{L}$ establishes the optimal trade-off relationship between the objectives of the design problem. From the Pareto solution, a number of insights can be drawn to assist with an effective design of MPC. First and foremost, the curve defines the best control performance that can be achieved for a range of computational capacity. Subsequently, a practitioner can select a particular design on the curve to achieve a specified performance, with the information on the minimum computational capacity of the hardware for real-time control implementation. By extension, the minimum cost for the hardware is can also be obtained. Other conclusions that can be made based on the Pareto front include the gradient of the curve which tells how much needs to be paid in terms of computational capacity for a unit increase in performance.

Lastly, although intended to be performed offline, note that to evaluate an objective point $\boldsymbol{\ell}(\mathbf{p})$ given the design parameter $\mathbf{p}$ a set of closed-loop simulations of the controlled plant needs to be performed. Multiple simulations are done to obtain a more representative measure of performance, for example, an average for a range of initial conditions, trajectories, or average performance under disturbance. This motivates the development of an optimiser that is both accurate and fast in finding the optimal solution to reduce the time required to obtain the Pareto front of interest.

3.8 Conclusions

A mathematical formulation of the proposed multi-objective design of MPC approach was presented in this chapter. Different elements of the design problem were formalised. First, the OCP that defines the MPC framework was formulated for the closed-loop control of a dynamic plant with its associated design objectives and parameters. The two pertaining design objectives were outlined, the first of which is control performance that is defined by the closed-loop value function, and the second being the required computational capacity measured by the utilisation number that defines a multiplier of the simulation hardware as the specified benchmark. Subsequently, coupled design parameters were identified, which are those that has dictates the design objectives. The parameters were further classified as either associated to the control algorithm or implementation aspect of the closed-loop system, the former being the focus of the investigation. Finally, the MOD-MPC approach was cast as a multi-objective optimisation problem, with an associated solution of the Pareto design choices that defines the optimal trade-off between the two objectives, providing useful insights for the practitioner. Obtaining the solution involves closed-loop simulations of the plant and can be time-consuming. Therefore, there is motivation for the use of an effective and efficient algorithm that can find the solution accurately and quickly, which will be discussed as part of the analytical results in the next chapter.

Chapter 4
Properties and solution of the MOD-MPC problem

A substantial proportion of this chapter has already appeared as articles in the journals Automatica and Control Engineering Practice [BKMM16, BMMK16a].

THE proposed multi-objective design of model-predictive control (MOD-MPC) approach has been defined as a multi-objective optimisation problem in Chapter 3. The novelty of such a scheme is recognised in the literature review, with current multi-objective design analyses existing almost exclusively for non-model-based controllers and without consideration of computational cost as a design objective [RMGNSB13]. In turn, a systematic solution technique for the MOD-MPC problem is lacking, leaving only general-purpose and off-the-shelf methods such as evolutionary algorithms [Deb01] available to use. Such methods might not be sufficient for the problem in hand as time-consuming closed-loop simulations are needed to evaluate objective points, even if it is intended to be done offline. A targeted optimiser algorithm that can solve the optimisation problem effectively and efficiently thus needs to be developed. This chapter acts on the established motivation by analytically investigating key properties of the optimisation problem, in particular the nature of the design objective functions and bounds on the optimal parameter set, for the purpose of developing a targeted algorithm illustrated by the design optimiser block in Fig. 3.1. Such an algorithm is subsequently presented and validated on two test cases involving missile pitch and submarine diesel engine control. The algorithm satisfies conditions prescribed for accuracy and speed of the optimiser, and compared against benchmarks from the literature to demonstrate its superior performance.

4.1 The dQ-OCP in dense representation

For the purpose of analysis, it is useful to recast the dQ-OCP (3.16) problem in a compact/dense representation. Although the decision variables of the dQ-OCP are given by $\mathbf{x}$ and $\mathbf{u}$, the relationship between the states and inputs is dictated

by the prediction model. Therefore, a specified input sequence is sufficient to define the entirety of the OCP decision variables. The problem then can be recast by writing the OCP in a dense QP form, where it is specified by the least possible number of optimisation variables and a uniform set of constraints (in this case, all inequalities). For the OCP recasting, let the optimisation variable be denoted by

$$\mathbf{z} := \mathbf{u} + \mathcal{R}\mathbf{x} + H^{-1}F^{\mathsf{T}}\mathsf{x}_i \tag{4.1}$$

where $\mathcal{R} := \operatorname{diag}(\overline{R}\,\overline{S}^{\mathsf{T}}, \ldots, \overline{R}\,\overline{S}^{\mathsf{T}}, \overline{R}_{\mathrm{r}}\overline{S}_{\mathrm{r}}^{\mathsf{T}})$, $\mathbf{z} =: (z_0, \ldots, z_N)$ and $\mathbf{z} \in \mathbb{R}^{Nn_{\mathrm{u}}}$. The composite cost weight

$$H := 2\left(\widetilde{\mathbf{B}}^{\mathsf{T}}\widetilde{\mathbf{Q}}\widetilde{\mathbf{B}} + \mathbf{R}\right) \tag{4.2}$$

is positive definite and invertible, as a consequence of the restriction of a positive semi-definite Q and Q_{t}, and a positive definite R when (3.4) is defined (Remark 3.1). Further, the following block matrices are defined,

$$L := 2\widetilde{\mathbf{A}}^{\mathsf{T}}\widetilde{\mathbf{Q}}\widetilde{\mathbf{A}} - FH^{-1}F^{\mathsf{T}} \qquad\qquad F := 2\widetilde{\mathbf{A}}^{\mathsf{T}}\widetilde{\mathbf{Q}}\widetilde{\mathbf{B}}$$

with block transition matrices

$$\widetilde{\mathbf{A}} := \begin{bmatrix} \underline{\widetilde{A}} \\ \widetilde{A}_{\mathrm{r}}\widetilde{A}^{N} \end{bmatrix} \qquad \widetilde{\mathbf{B}} := \begin{bmatrix} \underline{\widetilde{B}} \\ \widetilde{A}_{\mathrm{r}}\widetilde{A}^{N-1}\overline{B} & \widetilde{A}_{\mathrm{r}}\widetilde{A}^{N-2}\overline{B} & \cdots & \overline{B}_{\mathrm{r}} \end{bmatrix} \tag{4.3a}$$

$$\underline{\widetilde{A}} := \begin{bmatrix} \widetilde{A}^{0} \\ \widetilde{A}^{1} \\ \vdots \\ \widetilde{A}^{N} \end{bmatrix} \qquad \underline{\widetilde{B}} := \begin{bmatrix} 0 & \cdots & & \\ \overline{B} & & & \vdots \\ \widetilde{A}\overline{B} & \overline{B} & & 0 \\ \vdots & & \ddots & \vdots \\ \widetilde{A}^{N-1}\overline{B} & \widetilde{A}^{N-2}\overline{B} & & \overline{B} \end{bmatrix} \tag{4.3b}$$

and block cost weights

$$\widetilde{\mathbf{Q}} := \operatorname{diag}\left(\widetilde{Q}, \ldots, \widetilde{Q}, \widetilde{Q}_{\mathrm{r}}, Q_{\mathrm{t}}\right) \qquad\qquad \mathbf{R} := \operatorname{diag}\left(\overline{R}, \ldots, \overline{R}, \overline{R}_{\mathrm{r}}\right). \tag{4.4}$$

Here, the shifted matrices $\widetilde{A} := \overline{A} - \overline{B}\,\overline{R}^{-1}\overline{S}^{\mathsf{T}}$, $\widetilde{A}_{\mathrm{r}} := \overline{A}_{\mathrm{r}} - \overline{B}_{\mathrm{r}}\overline{R}_{\mathrm{r}}^{-1}\overline{S}_{\mathrm{r}}^{\mathsf{T}}$, $\widetilde{Q} := \overline{Q} - \overline{S}\,\overline{R}^{-1}\overline{S}^{\mathsf{T}}$ and $\widetilde{Q}_{\mathrm{r}} := \overline{Q}_{\mathrm{r}} - \overline{S}_{\mathrm{r}}\overline{R}_{\mathrm{r}}^{-1}\overline{S}_{\mathrm{r}}^{\mathsf{T}}$ account for the cross terms in the cost function (3.12a). Finally, let the matrices associated with the constraints be

$$Y := E + GH^{-1}F^{\mathsf{T}} \tag{4.5a}$$

$$G \coloneqq \begin{bmatrix} I - \mathcal{R}\underline{\widetilde{B}} \\ -(I - \mathcal{R}\underline{\widetilde{B}}) \\ \widetilde{\mathbf{B}} \\ -\widetilde{\mathbf{B}} \end{bmatrix} \qquad W \coloneqq \begin{bmatrix} \hat{\mathbf{u}} \\ -\check{\mathbf{u}} \\ \hat{\mathbf{x}} \\ -\check{\mathbf{x}} \end{bmatrix} \qquad E \coloneqq \begin{bmatrix} \mathcal{R}\underline{\widetilde{A}} \\ -\mathcal{R}\underline{\widetilde{A}} \\ -\widetilde{\mathbf{A}} \\ \widetilde{\mathbf{A}} \end{bmatrix}. \tag{4.5b}$$

Consequently, the OCP can be written in standard form as

$$\mathbf{z}^* \coloneqq \arg\min_{\mathbf{z}} J_z(\mathbf{z}, \mathbf{p}) \coloneqq \frac{1}{2}\|\mathbf{z}\|^2_{H(\mathbf{p})} + \frac{1}{2}\|\mathsf{x}_i\|^2_{L(\mathbf{p})} \tag{4.6a}$$

$$\text{s.t. } G(\mathbf{p})\mathbf{z} \leq W(\mathbf{p}) + Y(\mathbf{p})\mathsf{x}_i. \tag{4.6b}$$

The dependence of the solution $\mathbf{z}^*$ to $(\mathsf{x}_i, \mathbf{p})$ will be omitted unless needed for clarity. The associated value function of the OCP is

$$\begin{aligned} J_z^*(\mathsf{x}_i, \mathbf{p}) &\coloneqq J_z^*(\mathbf{z}^*(\mathsf{x}_i, \mathbf{p}), \mathbf{p}) \\ &= J^*(\mathsf{x}_i, \mathbf{p}) = \overline{J}^*(\mathsf{x}_i, \mathbf{p}) \text{ from (3.19)} \end{aligned}$$

The standard dense formulation is commonly used, e.g. in [BMDP02], except that $\mathbf{z}$ in (4.1) is modified slightly to accommodate for the cross terms associated with $\overline{S}$ and $\overline{S}_\mathrm{r}$ in (4.6a). Further, the residual prediction step changes the constituents of the block matrices.

4.2 Preliminary analysis

In this section, a preliminary analysis concerning the sensitivities of the objective functions with respect to the design parameters sampling time and prediction horizon length. In the investigations that follow, the restricting case (3.24) is not asserted, so that the prediction horizon length is generalised as a continuous parameter that is allowed to be a non-integer multiple of sampling time,

$$\begin{gathered} \mathbf{p} = (h, T) \in \mathcal{P}_\mathrm{c} \\ \text{where } \mathcal{P}_\mathrm{c} \coloneqq \{(h, T) : T \geq h\} \\ \subseteq \mathbb{R}_{>0} \times \mathbb{R}_{>0} \end{gathered}$$

Analysis will first be performed on the OCP in open-loop, in the absence of a controlled plant. The closed-loop system will be subsequently analysed with a linear plant, before the more realistic yet complex case with a non-linear plant is considered.

4.2.1 Underlying dQ-OCP properties

Differentiability of transition and cost matrices

The matrices $\overline{A}$, $\overline{B}, \overline{Q}, \overline{R}$, and $\overline{S}$ are differentiable with respect to h and independent of T. The Taylor series expansion for the perturbation $h^+ := h + \delta_h$ as $\delta_h \to 0$ is

$$\overline{A}^+ := \overline{A}(h^+) = \overline{A}(h) + \partial_h \overline{A}(h)\delta_h + \mathcal{O}(\delta_h^2)$$

and similarly for $\overline{B}, \overline{Q}, \overline{R}$ and $\overline{S}$. The matrices $\overline{A}_\mathrm{r}$, $\overline{B}_\mathrm{r}, \overline{Q}_\mathrm{r}, \overline{R}_\mathrm{r}$ and $\overline{S}_\mathrm{r}$ are differentiable with respect to h_r, thus differentiable with respect to (h, T), for all $h_\mathrm{r} \in (0, h)$ since $h_\mathrm{r} := T - Nh$.

Feasibility

Existence of a solution to the dQ-OCP (4.6) depends on the feasibility of the OCP, which in the context of designing the parameters of sampling time and prediction horizon length is given in the following definition.

Definition 4.1 (Feasible design set $\overline{\mathcal{P}}_\mathrm{F}^\mathrm{OL}(\mathsf{x}_i)$ of (4.6))**.** $\overline{\mathcal{P}}_\mathrm{F}^\mathrm{OL}(\mathsf{x}_i)$ contains all $\mathbf{p}$ for which the OCP (4.6) is feasible given a current state x_i [1].

Note that without state and input constraints (3.4d) such that $\mathcal{X} = \mathbb{R}^{n_\mathsf{x}}$ and $\mathcal{U} = \mathbb{R}^{n_\mathsf{u}}$, the OCP will always be feasible as long as the sampling time and prediction horizon length are chosen sensibly, that is, $\overline{\mathcal{P}}_\mathrm{F}^\mathrm{OL}(\mathsf{x}_i) = \mathcal{P}$, $\forall \mathsf{x}_i \in \mathbb{R}^{n_\mathsf{x}}$ under no state/input constraint.

Optimality of the OCP solution

The number of feasible solutions to the dQ-OCP is indefinite, given that the chosen design set is feasible according to Definition 4.1. Within the feasible solution set, the optimal solution of the dQ-OCP is well-defined, denoted by $\mathbf{z}^*$. The resulting optimal value function J_z^* is the global (and only local) minima, given the convexity of the QP [BV04]. Optimality of the solution is identified by the Karush-Kuhn-Tucker (KKT) conditions [WN99]. A solution $\mathbf{z}^*$ is the optimal solution to

[1] The complementary view is also useful; the feasible region $\mathcal{X}_\mathrm{F}(\mathbf{p})$ for a given $\mathbf{p}$ is a set of all current states x_i so that $\mathsf{x}_i \in \mathcal{X}_\mathrm{F}(\mathbf{p})$ implies that the OCP (4.6) is feasible.

the dQ-OCP iff

$$\exists \boldsymbol{\mu} : \begin{cases} -H\mathbf{z}^* = G^\mathsf{T}\boldsymbol{\mu} & \text{(4.7a)} \\ \boldsymbol{\mu} \geq 0 & \text{(4.7b)} \\ \mu_j c_j(\mathbf{z}^*) = 0 \qquad \forall j \in \{1, \ldots, n_c\}. & \text{(4.7c)} \end{cases}$$

The solution is $\mathbf{z}^* = -H^{-1}G^\mathsf{T}\boldsymbol{\mu}$. H is invertible as defined in (4.2). $\boldsymbol{\mu} \coloneqq (\mu_1, \ldots, \mu_{n_c})$ contains the Lagrange multipliers associated with the constraints denoted by $\mathbf{c}(\mathbf{z}) \leq 0$ where $\mathbf{c}(\mathbf{z}) \coloneqq G\mathbf{z} - W - Y\mathsf{x}_i$. c_j denotes the j^{th} constraint i.e. the j^{th} row of $c(\mathbf{z})$.

The active set $\mathcal{A}(\mathbf{z}^*) \coloneqq \{j | c_j(\mathbf{z}^*) = 0\}$ contains the indices of the active constraints at the solution $\mathbf{z}^*$. Let the accent $'$ denotes correspondence to active constraints, e.g. $\acute{\boldsymbol{\mu}} \coloneqq \{\mu_j | j \in \mathcal{A}(\mathbf{z}^*)\}$, $\acute{\boldsymbol{\mu}} \in \mathbb{R}^{n_a}$ and $n_a \coloneqq |\mathcal{A}|$ is the number of active constraints. From the complementarity condition (4.7c), elements of $\boldsymbol{\mu}$ for inactive constraints must be zero, and those for the active constraints are

$$\begin{aligned} 0 = \acute{\mathbf{c}} &= \acute{G}\mathbf{z}^* - \acute{W} - \acute{Y}\mathsf{x}_i \\ &= -\acute{G}H^{-1}\acute{G}^\mathsf{T}\acute{\boldsymbol{\mu}} - \acute{W} - \acute{Y}\mathsf{x}_i. \end{aligned}$$

Therefore,

$$\begin{aligned} \acute{\boldsymbol{\mu}} &= -\left(\acute{G}H^{-1}\acute{G}^\mathsf{T}\right)^{-1}\left(\acute{W} + \acute{Y}\mathsf{x}_i\right) \\ \boldsymbol{\mu} &= \mathcal{I}_{\mathcal{A}(\mathbf{z}^*)}\acute{\boldsymbol{\mu}}. \end{aligned} \tag{4.8}$$

The inverse in (4.8) exists iff $\acute{G}$ is full rank. $\mathcal{I}_{\mathcal{A}} \in \{0,1\}^{n_c \times n_a}$ is used to organise the Lagrange multipliers of the active constraint $\acute{\boldsymbol{\mu}}$ into respective rows of $\boldsymbol{\mu}$.

Uniqueness and non-degeneracy

Although the optimal value function J_z^* is unique for the dQ-OCP, the associated solution $\mathbf{z}^*$ might not be unique i.e. there might be more than one value of $\mathbf{z}^*$ that gives the optimal value function. It is therefore worthwhile to investigate the uniqueness of the solution $\mathbf{z}^*$. A sufficient condition for the uniqueness of the dQ-OCP solution is satisfaction of the linear independence constraint qualification (LICQ) [WN99], achieved if the active constraint gradients are linearly independent, i.e. $\acute{G}$ has full row rank, so that the solution (4.8) to the KKT conditions (4.7) is unique.

Definition 4.2 (Non-degenerate OCP)**.** An OCP (4.6) with solution $\mathbf{z}^*$ and active set $\mathcal{A}(\mathbf{z}^*)$ whose active constraint gradients $\acute{G}$ satisfies LICQ is defined as non-degenerate.

Non-degeneracy can, for example, be guaranteed by removing state constraints, leaving only constraints on the predictive input.

Lemma 4.1. *For a solution $\mathbf{z}^*$ of the OCP* (4.6)*, if $\mathcal{X} = \mathbb{R}^{n_\mathsf{x}}$ and $\underline{u} < \overline{u}$, then LICQ is satisfied.*

Proof. With no state constraints, G constitutes only the upper two row blocks in (4.5). Linear dependence comes from row pairs r and $r + (N+1)n_\mathsf{u}$ for $r \in \{1, \ldots, Nn_\mathsf{u}\}$, each with rows that are negative multiples of each other. These correspond to the upper and lower-bound for an input element that cannot both be active simultaneously. Therefore the active constraint gradients in the rows of $\acute{G}$ must be linearly independent. □

The exact conditions that guarantee non-degeneracy is outside the scope of the study, and the dQ-OCP is assumed to be non-degenerate, guaranteeing that regardless of the choice of parameters $\mathbf{p}$, the optimal solution is unique.

Assumption 4.1 (Non-degenerate OCP)**.** *The OCP* (3.4) *is non-degenerate so that its solution (u^*, x^*) is unique.*

Continuity of the OCP solution

The residual sampling time h_r ensures that the solution of the dQ-OCP remains continuous across changing N.

Lemma 4.2 (Continuity of $\mathbf{z}^*$ at $T/h \in \mathbb{N}_{>0}$)**.** *Consider the OCP* (4.6) *that is non-degenerate per Definition 4.3, given an $\mathsf{x}_i \in \mathbb{R}^{n_\mathsf{x}}$ such that $\mathbf{p} \in \overline{\mathcal{P}}_\mathrm{F}^\mathrm{OL}(\mathsf{x}_i) \neq \emptyset$. The unique optimal solution $\mathbf{p} \mapsto \mathbf{z}^*(\mathsf{x}_i, \mathbf{p})$ of the OCP is a continuous function of $\mathbf{p}$ at $T/h \in \mathbb{N}_{>0}$ i.e. at boundaries where N changes.*

Proof. An OCP (4.6) with N prediction steps and $h_\mathrm{r} = 0$ has the same solution $\mathbf{z}^*$ as that with $N-1$ steps and $h_\mathrm{r} = h$, relying on the non-degeneracy of the OCP (Assumption 4.1) hence uniqueness of $\mathbf{z}^*$. $\mathbf{z}^*$ has effectively the same dimension N for both cases. Hence, $\mathbf{z}^*$ must be continuous at the N to $(N-1)$ boundary. □

4.2.2 Open-loop analysis

The following analysis considers the optimal control at a given sampling instant in open-loop, without being closed in a feedback with a plant. The analysis is done around the sensitivity of the problem with respect to sampling time and prediction horizon length. Such an analysis forms the basis for the more comprehensive closed-loop study. In itself, the open-loop analysis extends investigations such as [BB14] that solely analyse sampling time.

Smoothness properties

To show smoothness properties, a perturbation $h^+ := h + \delta_h$ and/or $T^+ := T + \delta_T$ as $\delta_h \to 0$ and/or $\delta_T \to 0$ is performed. Consider when the perturbation does not change $N = \lfloor T/h \rfloor$. Let the perturbed optimal solution be $\mathbf{z}^+$, with perturbed Lagrange multipliers $\boldsymbol{\mu}^+$ and perturbed value function J_z^{*+}. To prove that J_z^* is differentiable, the differentiability of the Lagrange multipliers and dQ-OCP solution are first considered.

Lemma 4.3 (Differentiability of $\boldsymbol{\mu}$)**.** *The Lagrange multipliers $\boldsymbol{\mu}$ associated with an OCP* (4.6) *are differentiable with respect to $\mathbf{p}$ for a given N and satisfaction of Assumption 4.1.*

Proof. The proof is given in Appendix C.1. □

Lemma 4.4 (Continuity of $\mathbf{z}^*$)**.** *For an OCP* (4.6)*, the solution $\mathbf{z}^*$ is differentiable with respect to $\mathbf{p}$ for a given N and satisfaction of Assumption 4.1. Guarantees of differentiability are lost across changes in N.*

Proof. The proof is given in Appendix C.2. □

Knowing the differentiability of the Lagrange multipliers and solution to the dQ-OCP, the main result of the differentiability of the open-loop value function can be stated.

Theorem 4.1 (Continuity of J_z^*)**.** *Consider the OCP* (4.6)*, given an $\mathsf{x}_i \in \mathbb{R}^{n_x}$ such that $\mathbf{p} \in \overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{OCP}}(\mathsf{x}_i) \neq \emptyset$ and Assumption 4.1 is satisfied. The open-loop value function $\mathbf{p} \mapsto J_z^*(\mathsf{x}_i, \mathbf{p})$ is a*

- *continuous function of $\mathbf{p}$,*
- *differentiable function of $\mathbf{p}$ for a fixed $N = \lfloor h/T \rfloor$.*

Proof. Consider the value function $J_z^*(\mathsf{x}_i, h, T)$ of problem (4.6) at a solution $\mathbf{z}^*$, with $\boldsymbol{\mu}$ satisfying the KKT conditions (4.7). Consider a perturbation $h^+ := h + \delta_h$ as $\delta_h \to 0$ and $T^+ := T + \delta_T$ as $\delta_T \to 0$ that does not change $N = \lfloor T/h \rfloor$. From Lemma 4.4, and satisfaction of Assumption 4.1, the perturbed value function can be expressed as

$$\begin{aligned}
&J_z^*(\mathsf{x}_i, h^+, T^+) \\
&= J_z^+(\mathbf{z}^+) + \partial_h J_z(\mathbf{z}^+)\delta_h + \partial_T J_z(\mathbf{z}^+)\delta_T + \mathcal{O}(\delta_h^2, \delta_T^2) \\
&= \|\mathbf{z}^* + \partial_h z \delta_h + \partial_T z \delta_T\|^2_{H+\partial_h H\delta_h+\partial_T H\delta_T} + \|\mathsf{x}_i\|^2_{L+\partial_h L\delta_h+\partial_T L\delta_T} + \mathcal{O}(\delta_h^2, \delta_T^2) \\
&= J_z^*(\mathsf{x}_i, h, T) + \partial_h J_z^* \delta_h + \partial_T J_z^* \delta_T + \mathcal{O}(\delta_h^2, \delta_T^2)
\end{aligned}$$

for some finite $\partial_h J_z^*(\mathbf{p})$ and $\partial_T J_z^*(\mathbf{p})$. Hence, the value function J_z^* is differentiable, thus also continuous with respect to $\mathbf{p}$ for a fixed N. At boundaries where N changes, continuity of the value function is implied by Lemma 4.2. □

Numerical example

To illustrate the results, consider a harmonic oscillator

$$\dot{\mathsf{x}} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \mathsf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathsf{u}. \tag{4.9}$$

with $\omega = 1$. Following a number of trials on different plant models, the harmonic oscillator is chosen as it helps to best demonstrate the theoretical results numerically. The cost weights are chosen as

$$Q = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = 1 \tag{4.10}$$

and Q_t is chosen differently for the two examples considered, $Q_\mathrm{t} = 10Q$ for the first and $Q_\mathrm{t} = \mathbf{0}_{2\times 2}$ for the second.

If no state constraint is imposed, non-degeneracy is guaranteed (Lemma 4.1). An illustrative case is given in Fig. 4.1, where only the input is constrained to be $u \in [-1/4, 1/4]$. The open-loop value-function remains continuous and appears differentiable with changing active set and non-differentiable across changes in N, as per Theorem 4.1. Non-monotonicity with respect to $\mathbf{p}$ is observed, particularly more often at small sampling times.

The problem is degenerate if state constraints are imposed (in relation to Lemma 4.1). Solution to the KKT conditions is non-unique, inconsistent with

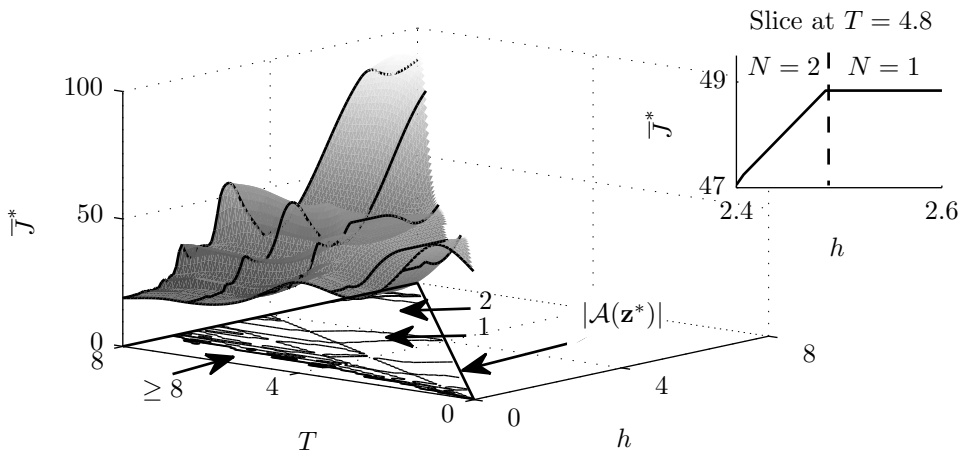

Figure 4.1: Open-loop value function $J_z^* = \overline{J}^*$ for the harmonic oscillator (4.9) with the dQ-OCP (4.6). $u \in [-1/4, 1/4]$, $\mathsf{x}_i = (1,1)$, $Q_\mathrm{t} = 10Q$. The contour of $|\mathcal{A}(\mathbf{z}^*)|$ is plotted on the z-plane.

(3.27). Consequently, continuity is not guaranteed anymore across active set changes, as seen in Fig. 4.2. Discontinuity happens at $h = \pi$, where the size of the active set changes to 2 from either 5 or 4. The loss of guarantee is consistent with the results in [BBBM08]. The figure also demonstrates the region associated with the infeasible design set $\mathbf{p} \notin \overline{\mathcal{P}}_\mathrm{F}^\mathrm{OCP}(\mathsf{x}_i)$.

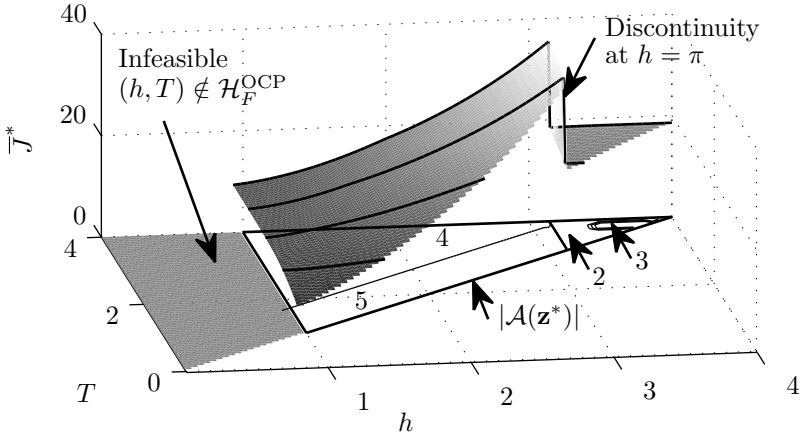

Figure 4.2: Open-loop value function $J_z^* = \overline{J}^*$ for the harmonic oscillator (4.9) with the dQ-OCP (4.6). $x \in [-1,1]$, $u \in [-1,1]$, $\mathsf{x}_i = (1,1)$, $Q_\mathrm{t} = \mathbf{0}_{2\times 2}$. The contour of $|\mathcal{A}(\mathbf{z}^*)|$ is plotted on the z-plane.

4.2.3 Closed-loop analysis with linear plant

The linear closed-loop system is defined by $\dot{\mathsf{x}}(t) = \mathsf{A}\mathsf{x} + \mathsf{B}\kappa(\mathsf{x}, \mathbf{p})$, $\forall t \in [ih, ih + h)$, $\forall i \in \mathbb{N}_{\geq 0}$, and its discrete-time equivalence

$$\mathsf{x}_{i+1} = \overline{\mathsf{A}}\mathsf{x}_i + \overline{\mathsf{B}}\kappa(\mathsf{x}_i, \mathbf{p}) \qquad \forall i \in \mathbb{N}_{\geq 0} \tag{4.11}$$

to follow the non-linear closed-loop system definition in (3.2). As in (3.17), $\overline{\mathsf{A}} := e^{\mathsf{A}h}$ and $\overline{\mathsf{B}} := \int_0^h e^{\mathsf{A}(h-\tau)} d\tau \, \mathsf{B}$. Since the plant is linear, the control law

$$\kappa(\mathsf{x}_i, \mathbf{p}) := u_0^* \tag{4.12}$$

is naturally obtained from the OCP with a linear prediction model (4.6) (derived from (3.16)) with $\overline{A} = \overline{\mathsf{A}}$, and $\overline{B} = \overline{\mathsf{B}}$.

The linear closed-loop value function is thus defined as follows

$$\overline{U}^*(\mathsf{x}_0, \mathbf{p}) := \sum_{i=0}^{\infty} \overline{U}(\mathsf{x}_i, \mathsf{u}_i, \mathbf{p}) \tag{4.13a}$$

$$\left.\begin{aligned} \text{s.t. } \mathsf{x}_{i+1} &= \overline{\mathsf{A}}\mathsf{x}_i + \overline{\mathsf{B}}\mathsf{u}_i \quad \forall i \in \mathbb{N}_{\geq 0} \\ \mathsf{u}_i &= \kappa(\mathsf{x}_i, \mathbf{p}). \end{aligned}\right\} \tag{4.13b}$$

The cost function is chosen, again, as a quadratic as per (3.22)

$$\overline{U}(\mathsf{x}_i, \mathsf{u}_i, \mathbf{p}) := \begin{bmatrix} \mathsf{x}_i \\ \mathsf{u}_i \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^{\mathsf{T}} & \overline{R} \end{bmatrix} \begin{bmatrix} \mathsf{x}_i \\ \mathsf{u}_i \end{bmatrix} \tag{4.14}$$

where the cost weighting matrices are as defined in (3.18) so that $\overline{U}(\mathsf{x}_i, \mathsf{u}_i, \mathbf{p}) = U(\mathsf{x}_i, \mathsf{u}_i)$. Note that the dependence of this cost function to u_i can be omitted when the full system (4.13) is considered, since $\mathsf{u}_i = \kappa(\mathsf{x}_i, \mathbf{p})$. To numerically calculate the value function $\overline{U}^*$, the closed-loop simulation must be performed in finite time $t \in [0, T_{\text{sim}} := N_{\text{sim}}h]$ so that the infinite series is truncated after $N_{\text{sim}} \in \mathbb{N}_{>0}$ steps.

Feasibility and stability of the closed-loop system

In closed-loop, the notion of feasibility (Definition 4.1) is extended to recursive feasibility. This requires control invariance, such that for an initial state x_0 the MPC law is feasible and keeps the system (4.11) feasible at all subsequent sampling steps i.e. $\mathsf{x}_{i+1} = \overline{\mathsf{A}}\mathsf{x}_i + \overline{\mathsf{B}}\kappa(\mathsf{x}_i, \mathbf{p}) \in \mathcal{X}_{\text{F}}$, $\forall i \in \mathbb{N}_{\geq 0}$ (see Footnote 1). This can be guaranteed with appropriately chosen design parameters h and T [May14].

Definition 4.3 (Feasible design set $\overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{CL}}$)**.** The feasible design set $\overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{CL}}(\mathsf{x}_0)$ contains all $\mathbf{p}$ such that the linear closed-loop system (4.13) is recursively feasible given an initial state x_0.

Remark 4.1. As in the open-loop case, under no state/input constraint, the feasible design set of the closed-loop system covers all choice of the design parameters as long as it is sensible so that $\overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{CL}}(\mathsf{x}_0) = \mathcal{P}, \forall \mathsf{x}_0 \in \mathbb{R}^{n_\mathsf{x}}$.

The closed-loop value function in (4.13a) is an infinite series. For it to be finite-valued, it has to be convergent, which can be guaranteed by stability of the closed-loop system.

Definition 4.4 (Stable design set $\overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}$)**.** The stable design set $\overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}(\mathsf{x}_0) \subseteq \overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{CL}}(\mathsf{x}_0)$ contains all $\mathbf{p}$ such that the linear closed-loop system (4.13) has a region of attraction containing the initial state x_0 as well as an origin that is reachable and exponentially stable.

Lemma 4.5 (Exponential stability)**.** *If* $(\overline{A}, \overline{B})$ *in the OCP* (3.16) *is reachable, and* $\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^{\mathsf{T}} & \overline{R} \end{bmatrix}$ *and* $\begin{bmatrix} \overline{Q}_r & \overline{S}_r \\ \overline{S}_r^{\mathsf{T}} & \overline{R}_r \end{bmatrix}$ *are positive semi-definite, then the origin of the closed-loop system* (4.13) *with an unconstrained OCP* (3.16a)–(3.16c) *is exponentially stable.*

Proof. The proof of the lemma follows Theorem 4.1 in [CGS84]. □

Remark 4.2. In Lemma 4.5, the reachability condition requires that (A, B) is reachable, and $h \neq 2\pi n / \Im(\lambda_i(A) - \lambda_j(A)), \forall i, j \in \{1, \ldots, n_x\}, i \neq j, n \in \mathbb{N}_{>0}$ [LSA71] when A has complex eigenvalues. Let $\mathcal{P}_{\mathrm{CR}}$ contain all of these 'critical' sampling times. The second condition regarding positive semi-definiteness is equivalent to having a positive semi-definite stage cost, which has been guaranteed by a positive semi-definite Q and positive definite R. It follows that $\overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}(\mathsf{x}_0) \subseteq \overline{\mathcal{P}}_{\mathrm{F}}^{\mathrm{CL}}(\mathsf{x}_0) \backslash \{\mathbf{p} | h \in \mathcal{P}_{\mathrm{C}}\}$. Stability of the closed-loop system (4.6) is discussed for example in [RM93].

Lemma 4.6. *For* $\mathbf{p} \in \overline{\mathcal{P}}_{S}^{CL}(\mathsf{x}_0)$*, the closed-loop value function* $U(\cdot, \cdot, \mathsf{x}_0)$ *in* (4.6) *is a convergent infinite series that is finite-valued.*

Proof. $\mathbf{p} \in \overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}(\mathsf{x}_0)$ implies the exponential stability of (4.13). Thus, the ratio test [Ste11] on (4.14) yields

$$\lim_{i \to \infty} \frac{\overline{U}(\mathsf{x}_{i+1}, \mathbf{p})}{\overline{U}(\mathsf{x}_i, \mathbf{p})} = \rho < 1.$$

Therefore, $\overline{U}^*$ is a convergent series and finite-valued in the limit $i \to \infty$. □

Smoothness properties

In control applications, the closed-loop value function is a true measure of performance. Smoothness properties of the closed-loop value function can be established following the results for the open-loop case.

Lemma 4.7 (Differentiability of state transition)**.** *The transition of the state* (4.13b) *of the closed-loop system* (4.13) *is differentiable with respect to* (h, T) *for a given* N*.*

Proof. Consider the state transition at time $i = 0$; $\mathsf{x}_1 = \overline{\mathsf{A}}\mathsf{x}_0 + \overline{\mathsf{B}}u_0^*(\mathsf{x}_0, h, T)$[2]. The OCP solution is differentiable with respect to (h, T) for a given N as per Lemma 4.4, hence also the optimal input. That is, the perturbation $h^+ := h + \delta_h$ as $\delta_h \to 0$ yields $u_0^{*+}(\mathsf{x}_i) = u_0^*(\mathsf{x}_i) + \partial_h u_0^*(\mathsf{x}_i)\delta_h + \mathcal{O}(\delta_h^2)$ and

$$\begin{aligned}\mathsf{x}_1^+ &= \left(\overline{\mathsf{A}} + \partial_h\overline{\mathsf{A}}\delta_h\right)\mathsf{x}_0 + \left(\overline{\mathsf{B}} + \partial_h\overline{\mathsf{B}}\delta_h\right)\left(u_0^*(\mathsf{x}_0) + \partial_h u_0^*(\mathsf{x}_0)\delta_h\right) + \mathcal{O}(\delta_h^2)\\ &= \mathsf{x}_1 + \partial_h\mathsf{x}_1\delta_h + \mathcal{O}(\delta_h^2).\end{aligned}$$

So, at each sampling instant after the initial, $\mathsf{x}_i^+ = \mathsf{x}_i + \partial_h\mathsf{x}_i\delta_h + \mathcal{O}(\delta_h^2)$, $\forall i \in \mathbb{N}_{>0}$. A perturbation on T affects the system similarly. □

Consider now the properties of the term $\overline{U}$ for the closed-loop value function $\overline{U}^*$ (4.13a), following Theorem 4.1.

Lemma 4.8 (Differentiability of $\overline{U}$)**.** *Consider the OCP* (3.16)*, given an* $\mathsf{x}_i \in \mathbb{R}^{n_x}$ *such that* $(h, T) \in \overline{\mathcal{P}}_F^{\text{OCP}}(\mathsf{x}_i) \neq \emptyset$ *and Assumption 4.1 is satisfied. The first element of the cost function* $(h, T) \mapsto \overline{U}(\mathsf{x}_i, h, T)$ *is differentiable with respect to* (h, T) *for a fixed* N*.*

Proof. For a perturbation $h^+ := h + \delta_h$ as $\delta_h \to 0$ which does not change N, $\overline{U}$ (4.14) is expressed as

$$\overline{U}^+ := \overline{U}(\mathsf{x}_i^+, h^+) = \left\| \begin{bmatrix} \mathsf{x}_i^+ \\ u_0^{*+}(\mathsf{x}_i) \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q}^+ & \overline{S}^+ \\ \overline{S}^{+\mathsf{T}} & \overline{R}^+ \end{bmatrix}}$$

for some perturbed cost weights $\overline{Q}^+$, $\overline{R}^+$ and $\overline{S}^+$, current state x_i^+, and optimal input $u_0^{*+}(\mathsf{x}_i)$. The cost weights are differentiable; $\overline{Q}^+ = \overline{Q} + \partial_h\overline{Q}\delta_h + \mathcal{O}(\delta_h^2)$

[2]Note that $\mathsf{u}_i = \kappa(\mathsf{x}_0, \mathbf{p}) = u_0^*(\mathsf{x}_0, \mathbf{p})$ as per (4.12).

etc. Lemma 4.7 states that the current state and optimal input are differentiable. Hence, the perturbed value of $\overline{U}$ can be expressed as

$$\begin{aligned}\overline{U}^{+} &= \left\| \begin{bmatrix} \mathsf{x}_i + \partial_h \mathsf{x}_i \delta_h \\ u_0^*(\mathsf{x}_i) + \partial_h u^* \delta_h \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q}+\partial_h \overline{Q}\delta_h & \overline{S}+\partial_h \overline{S}\delta_h \\ (\overline{S}+\partial_h \overline{S}\delta_h)^{\mathsf{T}} & \overline{R}+\partial_h \overline{R}\delta_h \end{bmatrix}} + \mathcal{O}(\delta_h^2) \\ &= \overline{U}(\mathsf{x}_i, h) + \partial_h \overline{U} \delta_h + \mathcal{O}(\delta_h^2).\end{aligned}$$

$\overline{U}$ is independent of T and thus the proof is complete. □

Theorem 4.2 (Continuity of $\overline{U}^*$)**.** *Consider the MPC system* (4.13)*, given an* $\mathsf{x}_0 \in \mathbb{R}^{n_x}$ *such that* $(h, T) \in \overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}(\mathsf{x}_0) \neq \emptyset$ *with an OCP* (3.16) *satisfying Assumption 4.1. The closed-loop value function* $(h, T) \mapsto \overline{U}^*(\mathsf{x}_0, h, T)$ *is a*

- *continuous function of* (h, T)*,*
- *differentiable function of* (h, T) *for a fixed* $N = \lfloor h/T \rfloor$ *as per* (3.15)*.*

Proof. The closed-loop value function (4.13a) is a sum of the terms $\overline{U}$ so that

$$\overline{U}^{*+} := \overline{U}^*(\mathsf{x}_0, h^+, T^+) = \overline{U}(\mathsf{x}_0, h^+, T^+) + \sum_{i=1}^{\infty} \overline{U}(\mathsf{x}_i^+, h^+, T^+).$$

For $(h, T) \in \overline{\mathcal{P}}_{\mathrm{S}}^{\mathrm{CL}}(\mathsf{x}_0)$, the summation series would be convergent (finite-valued) by Lemma 4.6. From Lemmas 4.7 and 4.8, $\overline{U}^{*+}$ can be expanded to yield

$$\begin{aligned}\overline{U}^{*+} &= \overline{U}(h^+, T^+, \mathsf{x}_0) + \\ &\qquad \overline{U}(h^+, T^+, \mathsf{x}_1 + \partial_h \mathsf{x}_1 \delta_h + \partial_T \mathsf{x}_1 \delta_T) + \\ &\qquad\quad \overline{U}(h^+, T^+, \mathsf{x}_2 + \partial_h \mathsf{x}_2 \delta_h + \partial_T \mathsf{x}_2 \delta_T) + \ldots + \\ &\qquad\qquad \overline{U}(h^+, T^+, \mathsf{x}_\infty + \partial_h \mathsf{x}_\infty \delta_h + \partial_T \mathsf{x}_\infty \delta_T) + \mathcal{O}(\delta_h^2, \delta_T^2) \\ &= \overline{U}^*(\mathsf{x}_0, h, T) + \partial_h \overline{U}^* \delta_h + \partial_T \overline{U}^* \delta_T + \mathcal{O}(\delta_h^2, \delta_T^2).\end{aligned}$$

for some finite $\partial_h \overline{U}^*(h, T)$ and $\partial_T \overline{U}^*(h, T)$. Therefore, the value function $\overline{U}^*$ is differentiable with respect to (h, T) for a fixed N. At boundaries where N changes, continuity of the value function is implied by Lemma 4.2. □

Numerical example

Fig. 4.3 plots the closed-loop value function for the harmonic oscillator (4.9) with cost weights (4.10) and $Q_{\mathsf{t}} = P_{\mathrm{DARE}}$. P_{DARE} is the solution of the discrete algebraic Riccati equation, chosen to emulate an infinite horizon cost. The function is continuous as per Theorem 4.2. Non-differentiability at changing N is observed.

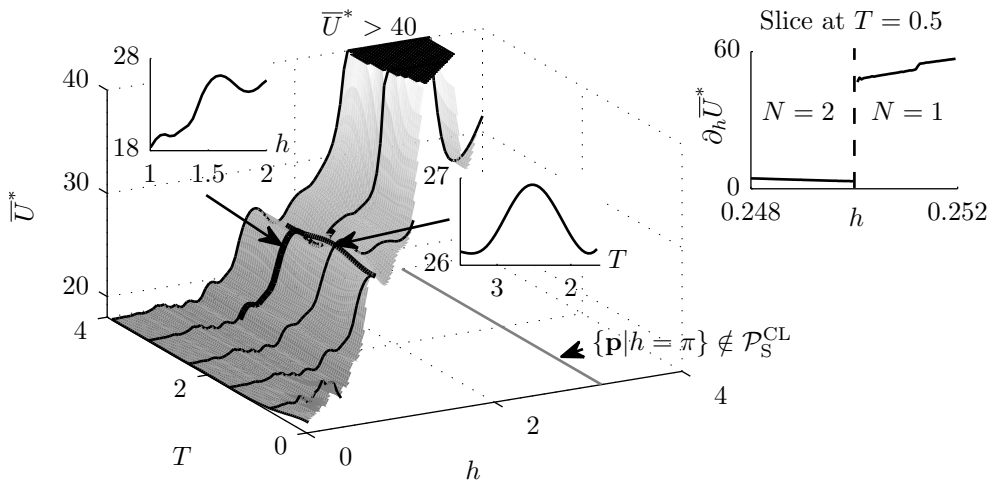

Figure 4.3: Closed-loop value function $\overline{U}^*$ for the harmonic oscillator (4.9). $|\mathsf{u}| \leq \frac{1}{4}$, $\mathsf{x}_0 = [1\ 1]^\mathsf{T}$, $Q_\mathrm{t} = P_\mathrm{DARE}$, $N_\mathrm{sim} = 250$. z-axis is truncated for clarity.

The value function becomes very large around $h = \pi$, as $\pi \in \mathcal{P}_C$ is a critical sampling time for the system (4.9) [LSA71]. The discrete-time plant $(\overline{\mathsf{A}}, \overline{\mathsf{B}})$ loses full controllability and is unstable, as $\{(h, T)|h = \pi\} \notin \overline{\mathcal{P}}_S^{CL}(\mathsf{x}_0)$ (Remark 4.2). At these points the value function $\overline{U}^*$ becomes unbounded and continuity is lost.

One might expect that longer prediction horizons and/or shorter sampling intervals are always desirable; a longer prediction horizon retrieves more future information while a shorter sampling interval gives better information resolution and more responsiveness, since control actions can be applied at a higher rate. Fig. 4.3 demonstrates that these expectations are however not necessarily upheld. It shows that there are multiple local minima on the surface with respect to both sampling interval and prediction horizon, particularly at smaller sampling times. The non-monotonicity is made clear by the subplots showing slices for a given h or T. This is also observed in the open-loop case with respect to h (Fig. 4.1).

4.3 Key properties of the MOD-MPC system

Having analysed the preliminary case involving a linear controlled plant, this section will consider the more relevant case where a non-linear plant is controlled in a closed-loop MPC system. This section will analyse the MOD-MPC system (3.28) to reveal its key attributes under the chosen design parameters defined in Section 3.6. These include the smoothness properties and bounds on the objective

and design parameters, establishing the underlying assumptions and guarantees that are useful for the subsequent development of a numerical optimiser for the problem. Based on the analytical foundation, a targeted solution method that is both effective and efficient can be appropriately developed.

4.3.1 Nature of the design objectives

Smoothness properties of the two objectives are fundamental in the theoretical formulation of the MOD-MPC problem. In particular monotonicity properties underlies the conclusion of the nature of the extrema of the optimisation, which is either associated with multiple local minima or a single minimum. Furthermore, monotonicity properties reveal the competing nature of the two objectives, in which one objective can only be made better off by compromising the other. Finally, the revealed continuity properties of the design objectives can be adapted for the efficiency of the developed optimiser.

Monotonicity

The model used for the required computational capacity η (3.27) has the following monotonicity property.

Proposition 4.1 (Monotonicity of η)**.** *Consider the required computational capacity $\eta(\cdot)$ in* (3.27) *and that Assumption 3.6 holds. For $\mathbf{p} := (h, N)$, $\mathbf{p} \mapsto \eta(\mathbf{p})$ is:*
- *monotonically decreasing with respect to h,*
- *monotonically increasing with respect to N.*

Proof. The monotonicity of η can be directly taken from the dependence of η to h and N as given in (3.27). □

On the other hand, the control performance as measured by the closed-loop value function V^* is non-monotonic.

Proposition 4.2 (Non-monotonicity of V^* with respect to h)**.** *Consider the control performance $V^*(\mathcal{X}_0, \cdot)$, $\mathcal{X}_0 \subseteq \mathcal{X}_S$ as in* (3.23) *and that Assumptions 3.3, 3.4, 3.8 and 4.1 hold. For $\mathbf{p} := (h, N)$, $h \mapsto V^*(\mathcal{X}_0, \mathbf{p})$ is generally non-monotonic. Specifically, the sets $\mathcal{H}_+$ and $\mathcal{H}_-$ are such that $h \mapsto V^*(\mathcal{X}_0, \mathbf{p})$ is monotonically increasing $\forall h \in \mathcal{H}_+$ and monotonically decreasing $\forall h \in \mathcal{H}_-$.*

Proof. Consider the closed-loop system[3]. For a fixed N, increasing h both desirably increases the prediction horizon length $T = Nh$ but also undesirably slowing down the sampling rate $1/h$ of the closed-loop system. At the limit $h \to 0$ ($T \to 0$), the system performs badly since the control prediction barely, if at all, captures any dynamics of the system. As h is increased, performance is improved, until some point where all important dynamics are captured. Increasing prediction length is no longer as influential as the delayed sampling rate, after which the performance is worsened with increasing h. Therefore, the value function $U(\mathsf{x}_0, (\cdot, N))$ is generally non-monotonic with respect to h. V is a linear combination of U and thus obeys the same properties. Consequently, there are two mutually exclusive sets $\mathcal{H}_+$ and $\mathcal{H}_-$, where $\mathcal{H}_+ \cup \mathcal{H}_- = \mathbb{R}_{>0}$, such that for all $h_a \in \mathcal{H}_+$ and $h_b \in \mathcal{H}_+$, if $h_b \geq h_a$ then $V(\mathcal{X}_0, (h_b, N)) \geq V(\mathcal{X}_0, (h_a, N))$. $\mathcal{H}_-$ is defined similarly. □

Proposition 4.3 (Non-monotonicity of V^* with respect to N)**.** *Consider the control performance $V^*(\mathcal{X}_0, \cdot)$, $\mathcal{X}_0 \subseteq \mathcal{X}_\mathsf{S}$ as in* (3.23) *and that Assumptions 3.3, 3.4, 3.8 and 4.1 hold. For $\mathbf{p} := (h, N)$, $N \mapsto V^*(\mathcal{X}_0, \mathbf{p})$ is generally non-monotonic. Specifically, the sets $\mathcal{N}_+$ and $\mathcal{N}_-$ are such that $N \mapsto V^*(\mathcal{X}_0, \mathbf{p})$ is monotonically increasing $\forall N \in \mathcal{N}_+$ and monotonically decreasing $\forall N \in \mathcal{N}_-$.*

Proof. Consider the closed-loop system[3]. For a fixed h, increasing N increases the prediction horizon length $T = Nh$. As a result more dynamics, as well as plant-model mismatch, are captured by the prediction. There is a trade-off balance between the two so that the value function $U(\mathsf{x}_0, (h, \cdot))$ is generally non-monotonic with respect to N. V is a linear combination of U and thus obeys the same properties. Consequently, there are two mutually exclusive sets $\mathcal{N}_+$ and $\mathcal{N}_-$, where $\mathcal{N}_+ \cup \mathcal{N}_- = \mathbb{N}_{>0}$, such that for all $N_a \in \mathcal{N}_-$ and $N_b \in \mathcal{N}_-$, if $N_b \geq N_a$ then $V(\mathcal{X}_0, (h, N_b)) \leq V(\mathcal{X}_0, (h, N_a))$. $\mathcal{N}_+$ is defined similarly. □

Remark 4.3. Propositions 4.2 and 4.3 are confirmed by the numerical observations in Section 4.2.

[3] with an OCP (3.16) satisfying Assumptions 3.3, 3.4, 3.8 and 4.1 with sampling time h, N prediction steps, a solution (x^*, u^*) and closed-loop value function $U^*(\mathsf{x}_0, (h, N))$, $\forall \mathsf{x}_0 \in \mathcal{X}_\mathsf{S}$ as in (3.21) .

Continuity and differentiability

The continuity of the solution of the OCP (4.6) (derived from (3.16)) with respect to $\mathbf{p}$ is described in the following.

Lemma 4.9 (Continuity of $\mathbf{u}^*$)**.** *Consider the OCP* (4.6) *satisfying Assumptions 3.3, 3.4, 3.8 and 4.1. The unique optimal solution* $\mathbf{p} \mapsto \mathbf{u}^*(\mathsf{x}_i, \mathbf{p})$ *of the OCP is differentiable with respect to h for a given N.*

Proof. The proof follows the continuity of $\mathbf{z}^*$ given in Lemma 4.4. From (4.1), $\mathbf{z}^* := \mathbf{u}^* + \mathcal{R}\mathbf{x}^* + H^{-1}F^{\mathsf{T}}\mathsf{x}_i$, therefore $\mathbf{u}^*$ follows the continuity properties of $\mathbf{z}^*$. □

Theorem 4.3 (Continuity of V^*)**.** *Consider the control performance* $V^*(\mathcal{X}_0, \mathbf{p})$, $\mathcal{X}_0 \subseteq \mathcal{X}_\mathsf{S}$ *as in* (3.23) *and that Assumptions 3.3, 3.4, 3.8 and 4.1 hold. If Assumption 3.1 holds, for* $\mathbf{p} := (h, N)$, $h \mapsto V^*(\mathbf{p})$ *is continuous with respect to h.*

Proof. Let $\mathsf{f}(\cdot,\cdot)$ satisfy Assumption 3.1. Let $\mathsf{z}(\cdot,\cdot)$ be the solution of $\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x},\mathsf{u})$. $\mathsf{u} \mapsto \mathsf{z}(t,\mathsf{u})$ is continuous with respect to u (Theorem 3.5 in [Kha02]). The control law $\mathsf{u}(\tau) = u_0^*(\mathsf{x}(t_i), \mathbf{p})$, $\forall \tau \in [ih, ih+h)$, $\forall i \in \mathbb{N}_{\geq 0}$ in (3.21d) where $\mathbf{p} := (h, N)$. From Lemma 4.9, u is continuous and differentiable with respect to h. This implies that the solution $h \mapsto \mathsf{z}(t, u_0^*(\mathsf{x}(t_i), (h, N)))$ is continuous with respect to h for a given N. $h \mapsto U^*(\mathsf{x}_0, (h, N))$, $\forall \mathsf{x}_0 \in \mathcal{X}_\mathsf{S}$ in (3.21) is thus continuous with respect to h for a given N and satisfaction of Assumptions 3.3, 3.4, 3.8 and 4.1. V^* is a linear combination of U^* and thus has the same monotonicity properties. □

The differentiability of the closed-loop value function can be described and is stated in the following.

Theorem 4.4 (Differentiability of V^*)**.** *Consider the control performance* $V^*(\mathcal{X}_0, \mathbf{p})$, $\mathcal{X}_0 \subseteq \mathcal{X}_\mathsf{S}$ *as in* (3.23) *and that Assumptions 3.3, 3.4, 3.8 and 4.1 hold. If Assumptions 3.1 and 3.2 hold, for* $\mathbf{p} := (h, N)$, $\mathbf{p} \mapsto V^*(\mathbf{p})$ *is differentiable with respect to h for a given N.*

Proof. Let $\mathsf{f}(\cdot,\cdot)$ satisfy Assumptions 3.1 and 3.2. Let $\mathsf{z}(\cdot,\cdot)$ be the solution of $\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x},\mathsf{u})$. $\mathsf{u} \mapsto \mathsf{z}(t,\mathsf{u})$ is differentiable with respect to u (Theorem 3.5 and Section 3.3 in [Kha02]). The control law $\mathsf{u}(\tau) = u_0^*(\mathsf{x}(t_i), \mathbf{p})$, $\forall \tau \in [ih, ih+h)$, $\forall i \in \mathbb{N}_{\geq 0}$ in (3.21d) where $\mathbf{p} := (h, N)$. From Lemma 4.9, u is continuous and differentiable

with respect to h. This implies that the solution $h \mapsto \mathsf{z}(t, u_0^*(\mathsf{x}(t_i), (h, N)))$ is differentiable with respect to h for a given N. $h \mapsto U^*(\mathsf{x}_0, (h, N)), \forall \mathsf{x}_0 \in \mathcal{X}_\mathrm{S}$ in (3.21) is thus differentiable with respect to h for a given N and satisfaction of Assumptions 3.3, 3.4, 3.8 and 4.1. V^* is a linear combination of U^* and thus has the same monotonicity properties. □

Competing nature

The competing nature of a pair of functions that are both to be minimised (or maximised) is defined in Definition 1.6. In the multi-objective design of MPC, both the closed-loop value function V^* and required computational capacity η are to be minimised. Based on Proposition 4.1, as well as Propositions 4.2 and 4.3, the two design objectives of control performance and required computational capacity are competing. This is detailed in the following.

Lemma 4.10 (Competing design objectives)**.** *The objective functions $V^*(\mathcal{X}_0, \cdot)$, $\mathcal{X}_0 \subseteq \mathcal{X}_\mathrm{S}$, and $\eta(\cdot)$ are competing as per Definition 1.6 within the design parameter set $\mathcal{P}_\mathrm{C} = (\mathcal{H}_+ \times \mathcal{N}_-)$ from Proposition 4.1, Propositions 4.2 and 4.3, and given that Assumptions 3.3, 3.4, 3.6–3.8 and 4.1 hold.*

Proof. Consider Proposition 4.1, Propositions 4.2 and 4.3, and satisfaction of Assumptions 3.3, 3.4, 3.6–3.8 and 4.1. η is monotonically decreasing with respect to h and increasing with N. V^* is non-monotonic with respect to h and N, and there exist a set such that V^* is increasing with respect to h and decreasing with respect to N, given by $\mathcal{P}_\mathrm{C} = (\mathcal{H}_+ \times \mathcal{N}_-)$ from Propositions 4.2 and 4.3. Within this set, $\forall \mathbf{p} \in \mathcal{P}_\mathrm{C}$, $\mathbf{p} \mapsto V^*(\mathbf{p})$ is monotonically increasing whilst $\mathbf{p} \mapsto \eta(\mathbf{p})$ is monotonically decreasing, or vice versa. □

The search space $\mathcal{P}_\mathrm{s}$ is assumed to intersect with $\mathcal{P}_\mathrm{C}$ so that the MOD-MPC solution $\mathcal{P}_\bullet$ exist. Associated with the solution is the Pareto front $\mathcal{L}$ in (3.29) which consists of Pareto optimal points as defined in Definition 1.5, each of which is a Pareto design choice $\mathbf{p} \in \mathcal{P}_\bullet \subseteq \mathcal{P}_\mathrm{s}$.

Assumption 4.2. $\mathcal{P}_\bullet = \mathcal{P}_\mathrm{s} \cap \mathcal{P}_\mathrm{C} \neq \emptyset$.

Further to its Pareto optimality, the quality of a point can be specified by its rank, such that all Pareto optimal points in a given set of points have a rank of 1.

The Pareto optimal points in the set that excludes points with rank 1 have rank 2, and so on.

Definition 4.5 (Rank)**.** Given a countable set of points $\mathcal{P}$, if a point j is Pareto optimal then its rank $r_j = 1$. Subsequently, a point j has rank $r_j = \rho$ if it is Pareto optimal in $\mathcal{P} \setminus \mathcal{P}_{\rho-1}$ where $\mathcal{P}_\rho$ is the set of all points with rank $r \leq \rho$.

4.3.2 Bounds on the optimal parameter set

The first bounding of the design parameter comes from the fact that it is numerically impractical to search the open set $\mathcal{P}$ in (3.25). Hence, the search space $\mathcal{P}_\mathrm{s}$ in (3.28) must be a closed set $\mathcal{P}_\mathrm{s} \subset \mathcal{P}$ that is able to be practically searched to find the Pareto design set $\mathcal{P}_\bullet$. Next, an assumption on the Pareto design set can be made based on the nature of the design problem.

Assumption 4.3 (upper-bound on h)**.** *For a given $N > 1$, the Pareto design set is upper-bounded by $\hat{h}$. This bound is defined by the notion that $\exists \hat{h} \in \mathcal{P}_\mathrm{s}$ such that $\forall h > \hat{h}$, $\boldsymbol{\ell}((h, N-1)) \prec \boldsymbol{\ell}((h, N))$.*

Remark 4.4. Assumption 4.3 comes from the fact that as the sampling time h is increased for a given number of prediction steps N, the competitive effect of reducing the required computing capacity will be diminished and surpassed by the deterioration in control performance as a result of more infrequent sampling. At this point, using fewer prediction steps, e.g. $N-1$, would reduce the capacity required more competitively than deteriorating performance. The opposite is also true; as the sampling time is shortened, the effect of increasing control performance due to a finer sampling would eventually be overtaken by an increase in computing capacity required, after which increasing N, to e.g. $N+1$, would be more competitive in increasing performance.

Assumption 4.4 (Lower-bound on h)**.** *For a given $N \geq 1$, the Pareto design set is lower-bounded by $\check{h}$. This bound is defined by the notion that $\exists \check{h} \in \mathcal{P}_\mathrm{s}$ such that $\forall h < \check{h}$, $\boldsymbol{\ell}((h, N+1)) \prec \boldsymbol{\ell}((h, N))$.*

Based on the bounds on the sampling time and the smoothness properties of the design objectives defined in the earlier subsection, bounds can be specified for $\mathcal{P}_\bullet$.

Theorem 4.5 (Bound on $\mathcal{P}_\bullet$)**.** *Consider a rectangular search space $\mathcal{P}_\mathrm{s} = \{(h, N) : h \in [\underline{h}, \overline{h}]$ and $N \in [\underline{N}, \overline{N}]\}$ and that Assumptions 4.3 and 4.4 is satisfied. The Pareto design set $\mathcal{P}_\bullet$ is contained within the band $\mathcal{P}_\mathrm{b} := \{\mathbf{p} : h \leq m_1 N + h_1$ and $h \geq m_2 N + h_2\}$ for some negative gradients $m_i \in \mathbb{R}_{<0}$ and constants h_i, $i \in \{1, 2\}$.*

Proof. Consider a Pareto design set $\mathcal{P}_\bullet$ in a rectangular search space $\mathcal{P}_\mathrm{s} = \{(h, N) : h \in [\underline{h}, \overline{h}]$ and $N \in [\underline{N}, \overline{N}]\}$. For a given number of prediction steps $N > 1$, let the sampling times corresponding to the Pareto design set have an upper-bound from Assumption 4.3 denoted $\hat{h}_N$. Paraphrasing the assumption, $\exists h$ such that $\boldsymbol{\ell}((h, N-1)) \prec \boldsymbol{\ell}((h, N))$, $\forall h > \hat{h}_N$ for the given upper-bound $\hat{h}_N$. Consequently, there must exist an upper-bound associated with $N-1$ prediction steps $\hat{h}_{N-1}$ that is larger than the upper-bound $\hat{h}_N$, giving $\hat{h}_N < \hat{h}_{N-1}$. Therefore, the Pareto design set can be upper-bounded by a line of negative gradient with respect to N. An opposite notion can be made using Assumption 4.4 to form a lower-bound with a negative gradient. This gives a bound in the form of a band $\mathcal{P}_\mathrm{b}$ as in the theorem. □

4.4 Numerical solution of the MOD-MPC problem

4.4.1 Effective and efficient optimiser characteristics

Analysis of the key properties of the system results in a number of characteristics required by a proposed numerical optimiser used to solve the MOD-MPC problem (3.28) accurately and quickly, as summarised below.

Condition 4.1 (Convergent)**.** Lemma 4.10 implies that there is a Pareto optimal design set $\mathcal{P}_\bullet$ for a given search space $\mathcal{P}_\mathrm{s}$ associated with the trade-off of the competing objectives. The optimiser should be able to effectively find $\mathcal{P}_\bullet$ with certain guarantees.

Condition 4.1a (Global)**.** Propositions 4.1–4.3 define that the required computational capacity η is monotonic and that the value function V^* is non-monotonic. The optimiser needs to search globally and handle the many local optima on the objective surface.

Condition 4.1b (Able to handle discrete parameters)**.** The optimiser must be able to handle discrete design parameters defined in (3.25).

Remark 4.5. Conditions 4.1a and 4.1b are necessary conditions for Condition 4.1 to be fulfilled. Satisfaction of these two is not always sufficient to satisfy Condition 4.1.

Condition 4.1 (necessarily with Conditions 4.1a and 4.1b) is a sufficient condition for a numerical optimiser to be accurate (convergent to the solution) for the MOD-MPC problem. Additional features are necessary for the optimiser to converge quickly and efficiently.

Condition 4.2 (Continuous)**.** The optimiser could rely on Theorems 4.3 and 4.4 that define the continuity/differentiability of the value function V^* based on the knowledge of $\mathsf{f}(\cdot,\cdot)$.

Condition 4.3 (Focussed)**.** Theorem 4.5 states that for a rectangular search space, the Pareto optimal design set is located within a specific space defined as a band. As a consequence, the optimiser should be able to focus its search within the band and omit any ineffectual space.

Remark 4.6. Conditions 4.2 and 4.3 are sufficient for the optimiser to be efficient and performs better than a general-purpose optimiser.

4.4.2 Dividing Triangles (DITRI) optimiser

Having now specified the sufficient conditions for an effective and efficient optimiser, a specialised algorithm satisfying all the conditions can be proposed for the MOD-MPC problem. The algorithm is based on Lipschitzian optimisation [JPS93], denoted *Dividing Triangles* (DITRI).

Projection of bounds

The principles of Lipschitzian optimisation are outlined in Algorithm 1. In each iteration, given a set of point(s) P, a point $\mathbf{p}_j \in P$ is *potentially optimal* if its *projected* lower-bound of the (minimised) objective within the associated search space must be equal or better than all points in P. The search space of all potentially optimal points will be partitioned into smaller divisions and a new point is the evaluated in each subdivision.

The bound *projection* must be consistent throughout, dictated by a constant referred to as the Lipschitz constant, hence the name of the algorithm. The projected bound $L(d_j)$ of an evaluated objective $\ell(\mathbf{p}_j) \in \mathbb{R}^1$ in a search space of size

Algorithm 1 Generic Lipschitzian optimisation

Require: Search space bounds
1: Evaluate initial point(s)
2: **repeat**
3: Find a set O of potentially optimal points
4: **for all** $o \in O$ **do**
5: Evaluate new points based on the search space division of potentially optimal point o
6: **end for**
7: **until** iteration or evaluation count limit is reached

d_j is defined as $L(d_j) = \ell(\mathbf{p}_j) - K_\mathrm{L} d_j$ for some Lipschitz constant $K_\mathrm{L} > 0$. The projection relies on the continuity of the objective to comply with Condition 4.2. For a point $\mathbf{p}_j$, the bound is better (smaller) if it has a smaller objective $\ell(\mathbf{p}_j)$ and/or a bigger search space size d_j. The point j is potentially optimal if $L(d_j) \leq L(d_i)$ for all $i \in P$, that is

$$\ell(\mathbf{p}_j) - K_\mathrm{L} d_j \leq \ell(\mathbf{p}_i) - K_\mathrm{L} d_i, \ \forall i \in P \text{ for a given } K_\mathrm{L} > 0. \tag{4.15}$$

Potential optimality classification

Potential optimality of a design choice is classified by its Pareto optimality (Definition 1.5). More precisely, the classification is based on the rank r of the point (Definition 4.5). The specification of the Lipschitz constant is tightened from $K_\mathrm{L} > 0$ to $K_\mathrm{L} = \varepsilon$ where ε is a very small positive number, giving

$$r_j - K_\mathrm{L} d_j \leq r_j - K_\mathrm{L} d_i, \ \forall i \in P, \ K_\mathrm{L} = \varepsilon. \tag{4.16}$$

This criterion helps to quickly localise regions of optimal solutions, ultimately allowing for a more efficient convergence to comply with Condition 4.3.

Potential optimality selection can be intuitively illustrated on an f-d plot (Fig. 4.4). A point j satisfies (4.15) if there is a line intersecting the point with a gradient $K_\mathrm{L} > 0$ such that all other points lie above the line in f-d coordinates. Consequently, all potentially optimal points lie on the lower right edge of the convex hull of the points. The tightened requirement to $K_\mathrm{L} = \varepsilon$ in (4.16) effectively means that only points with the lowest objective values (rank) are chosen to be potentially optimal.

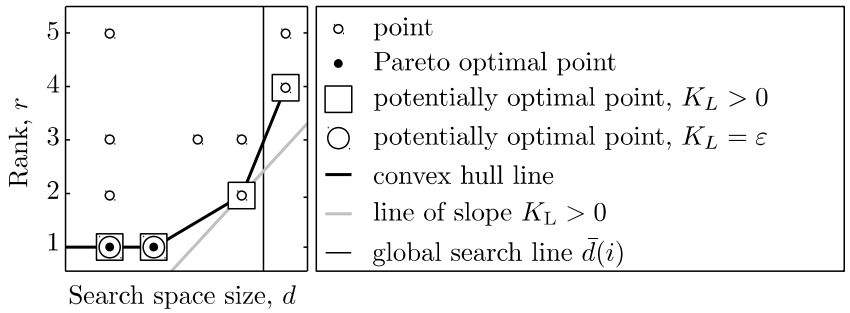

Figure 4.4: Potential optimality selection.

In addition to the criterion (4.16) to classify a point j as potentially optimal, the criterion

$$d_j > \overline{d}(i) \tag{4.17}$$

is used to guarantee convergence in Condition 4.1. $\overline{d}(i)$ is monotonically decreasing with i and $\lim_{i\to\infty} \overline{d}(i) = 0$. This asserts that search spaces that are relatively much bigger are divided, effectively guaranteeing a global search. Consequently, all the search space divisions will eventually be divided regardless of satisfaction of the condition (4.16).

Search space normalisation, bounds, division and size

The search space is bounded rectangularly, specified by $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$ containing the lower- and upper-bounds for each design parameters. A normalised point $\mathbf{c}$ is defined as

$$\begin{aligned} \mathbf{c} &= (\mathbf{p} - \underline{\mathbf{p}}) \oslash (\overline{\mathbf{p}} - \underline{\mathbf{p}}), \\ \text{so that } \mathbf{p} &= \underline{\mathbf{p}} + (\overline{\mathbf{p}} - \underline{\mathbf{p}}) \otimes \mathbf{c}. \end{aligned} \tag{4.18}$$

Theorem 4.5 allows for improving search efficiency by focusing on the band $\mathcal{P}_\mathrm{b}$ defined in the theorem. To efficiently locate the specified band, the search will be simplex-based (triangular) as illustrated in Fig. 4.5. In each iteration, every potentially optimal simplex is divided to form two simplices of equal size. The size measure d used in criteria (4.16) and (4.17) is the longest distance from the centre to the vertices of the simplex.

The efficiency of a simplex-based search comes from the fact a simplex is the basic polytope in any n-dimension. The approach is contrasted to the classical

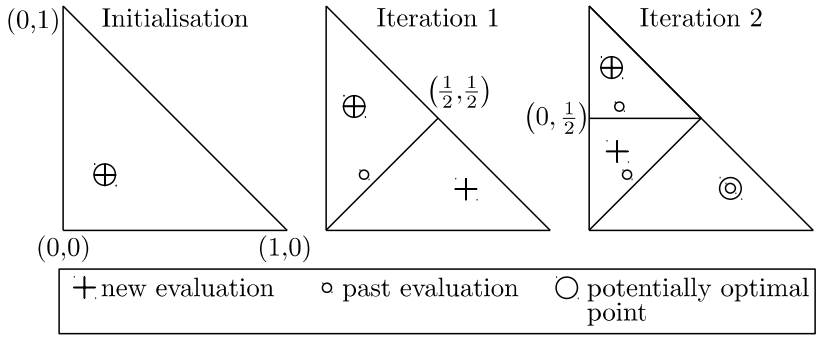

Figure 4.5: Illustration of simplex division over two iterations.

implementation of Lipschitzian optimisation whereby hyper-rectangles are used (Dividing Rectangles, or DIRECT) [Jon01]. The efficiency of DITRI is demonstrated in Fig. 4.6. DITRI does the minimum evaluations (two) per iteration and thus can adjust the search location more efficiently compared to DIRECT (up to four). For a given limit on evaluation count, DITRI can more efficiently locate the optimal regions $\mathcal{P}_\mathrm{b}$ in Theorem 4.5. This complies with Condition 4.3 for the proposed algorithm.

Evaluation point location

Point evaluation within a simplex is determined stochastically instead of (deterministically) at the centre. The uniform sampling is such that the expected value

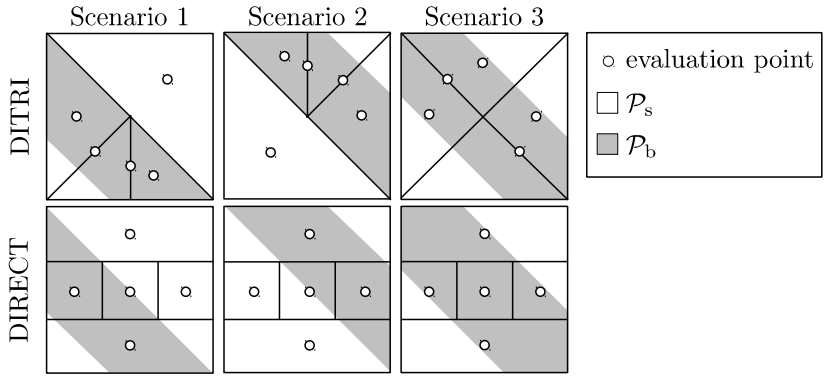

Figure 4.6: Comparison between the more adaptable and effective simplex-based search and rectangular search. Each evaluation point is placed in the middle of the search space. $\mathcal{P}_\mathrm{b}$ is defined in Theorem 4.5.

for the evaluation point chosen is at the centre of the simplex,

$$\mathbf{c} = \mathbf{s}_{n_p+1} \tag{4.19}$$

where $\mathbf{s}_1 := b_1$, $\mathbf{s}_i := b_i + \mathsf{U}[0,1]^{1/(i-1)}(\mathbf{s}_{i-1} - b_i)$, $\forall i \in \{2, \ldots, n_p + 1\}$, and b_i defines the coordinates of the i^{th} simplex vertex.

The approach of a random evaluation point allows a faster convergence rate on average than that achieved by a deterministic evaluation. This is possible because the number of instances when a random sample is better placed than the midpoint is on average equal to the instant when it is worse placed. At the case when a random sample is worse placed, its effect would be dominated by the better placed sample evaluations and diminished at subsequent iterations.

Finally, integer-valued design parameters are handled simply by shifting the design parameter p_i to the nearest integer, or to the immediate larger integer in the case of a half-integer, for any integer-valued design parameter i. This fulfils condition 4.1b. For the MOD-MPC problem in (3.28), the number of prediction steps N is the relevant integer-valued design parameter.

DITRI algorithm

The detailed outline of DITRI is given in Algorithm 2.

The choice of a Lipschitzian approach addresses Conditions 4.1a and 4.1b necessary for the optimiser to be convergent. Lipschitzian optimisation is gradient-free and is built for a global search, therefore addressing condition 4.1a. The method directly handles discrete design parameters specified in condition 4.1b. As a whole, the proposed optimiser is guaranteed to converge and satisfies Condition 4.1, making it an accurate optimiser that can effectively find the solution (Pareto design set) $\mathcal{P}_\bullet$ of the MOD-MPC problem (3.28) in a given search space.

Theorem 4.6 (Convergent DITRI)**.** *Consider a multi-objective optimal design problem with objectives $\mathbf{p} \mapsto \boldsymbol{\ell}(\mathbf{p})$ for $\mathbf{p} \in \mathcal{P}_\text{s}$ where $\mathcal{P}_\text{s}$ is a finite search space. Let the solution of the problem be $\mathcal{P}_\bullet \subseteq \mathcal{P}_\text{s}$. Also let $\{\mathcal{S}_i\}_{i=0}^{\bar{i}}$ for some indexing variable i be a sequence of solutions generated from a given initialisation $\mathcal{S}_0$. DITRI, described in Algorithm 2, is a convergent algorithm such that $\lim_{i \to i^\text{c}} S_i = \mathcal{P}_\bullet$ for some $i^\text{c} < \infty$.*

Proof. Let $P(i^\text{it})$ be the set containing all the evaluated points at iteration number i^it in Algorithm 2. After some i_+ steps ahead, every point in $P(i^\text{it})$ would eventually be classified as potentially optimal either via criterion (4.16) given its rank,

or criterion (4.17) given that each search space size d_j for all $j \in \{1, \ldots, |P(i^{\text{it}})|\}$ would become smaller than $\overline{d}(i^{\text{it}} + i_+)$. Each potential optimal search space will be divided and at least 1 new point will be evaluated after each division so that $P(i^{\text{it}} + i_+) \supset P(i^{\text{it}})$. Consequently, as $i_{\text{it}} \to \infty$, the algorithm would search the space $\mathcal{P}_{\text{s}}$ entirely i.e. $\lim_{i^{\text{it}} \to \infty} P(i^{\text{it}}) = \mathcal{P}_{\text{s}}$. Any point $\mathbf{p} \in \mathcal{P}_\bullet \subseteq \mathcal{P}_{\text{s}}$ in the Pareto design set will be evaluated so that $\lim_{i^{\text{it}} \to \infty} S_{i^{\text{it}}} = \mathcal{P}_\bullet$. □

Algorithm 2 DITRI

Require: Design parameter bounds $(\underline{\mathbf{p}}, \overline{\mathbf{p}})$, maximum evaluation count $\overline{i^{\text{ev}}}$ and maximum iteration count $\overline{i^{\text{it}}}$

$i^{\text{it}} \leftarrow 0$, $i^{\text{ev}} \leftarrow 0$
Let $\mathbf{\Phi}$ contain all evaluated points
Evaluate the two initial points,
$\mathbf{b}_1 = \{(1,0),(0,1),(0,0)\}$ and $\mathbf{b}_2 = \{(1,0),(0,1),(1,1)\}$, or
$\mathbf{b}_1 = \{(0,0),(1,1),(1,0)\}$ and $\mathbf{b}_2 = \{(0,0),(1,1),(0,1)\}$
for the two initial points $i = 1$ and 2 **do**
    Set normalised vertices of initial point $\mathbf{b}_i$
    Evaluate point $\ell(\mathbf{p}_i)$ in $\mathbf{b}_i$ using (4.18) and (4.19)
    Record initial point in $\mathbf{\Phi}$
    $i^{\text{ev}} \leftarrow i^{\text{ev}} + 1$
end for
$\mathcal{S}_0 \leftarrow \{\mathbf{p}_1, \mathbf{p}_2\}$
loop
    Find set O of potentially optimal points in $\mathbf{\Phi}$ based on (4.16) and (4.17)
    for all $o \in O$ **do**
        Divide simplex $\mathbf{b}_o$ to obtain simplices $\mathbf{b}_1^+$, $\mathbf{b}_2^+$
        for both $j = 1$ and 2 **do**
            Evaluate point $\ell(\mathbf{p}^+)$ in $\mathbf{b}_j^+$ using (4.18) and (4.19)
            Record new point in $\mathbf{\Phi}$
            $i^{\text{ev}} \leftarrow i^{\text{ev}} + 1$
            Update $\mathcal{S}_{i^{\text{ev}}}$ to be all Pareto points in $\mathbf{\Phi}$
            if $i^{\text{ev}} \geq \overline{i^{\text{ev}}}$ **then** terminate algorithm
        end for
    end for
    $i^{\text{it}} \leftarrow i^{\text{it}} + 1$
    if $i \geq \overline{i^{\text{it}}}$ **then** terminate algorithm
end loop

As well as being accurate, DITRI is developed to be an efficient optimiser by using projection of bounds. This assumes continuity in the objective function and satisfies Condition 4.2. Finally, DITRI conducts a focussed search of a given search space via a clever choice in potential optimality classification, search space division and evaluation point location within a given search subdivision as outlined throughout the subsection. Such a targeted approach fulfils Condition 4.3.

4.5 Numerical illustrations

To demonstrate the workings of the MOD-MPC approach, two test cases based on real-world examples are investigated. In these cases, Q and R are chosen accordingly for each case. The terminal cost is weighted by $Q_t = P_{ARE}$, where P_{ARE} is the solution of the algebraic Riccati equation for the simulated plant. The OCP is represented as a sparse QP. The global search criterion (4.17) is set as

$$\bar{d}(i) := \sqrt{\frac{5}{9}}^{i/8}$$

so that no search space is larger than the simplex equivalent to $i/8$ divisions from the initial simplex.

In the form (4.6), the resulting QP is classified as a dense program due to the density and tallness of the constraint matrices (4.5), that is, most of the constraint matrix elements are non-zero and that the number of optimisation variables is less than the number of constraints. Although useful analytically, it can be numerically less advantageous than a sparse representation which sets the states and inputs $(\mathbf{x}, \mathbf{u})$ explicitly as optimisation variables. A common example of such a formulation is given in Appendix A.1 which is used when the OCP is numerically solved to generate the results.

Intel® Core™i7-3770 Processor	
Cores (available for simulation)	4 (1)
Cycle frequency/core	3.4-3.9 GHz
Operations/cycle	8
FLOP/s (available for simulation)	$\sim$109-125 ($\sim$27-31) $\times 10^9$

Table 4.1: Simulation hardware specifications [Int15].

The OCP in sparse representation is solved using the interior point (barrier) method from the Gurobi solver in MATLAB [Gur13] with all tolerance values set as default. The specifications of the simulation hardware used are given in Table 4.1. Fig. 4.7 shows a representative result for the relationship between the solution time and number of prediction steps N for a range of sampling time h. It is shown that the solution time is generally increasing with N, verifying Assumption 3.6, and very weakly, if at all, correlated to h to verify Assumption 3.7. From the obtained data, the relationship for the chosen algorithm and QP form is mostly linear. The solution time upper-bound ζ in (3.26) is modelled as a linear

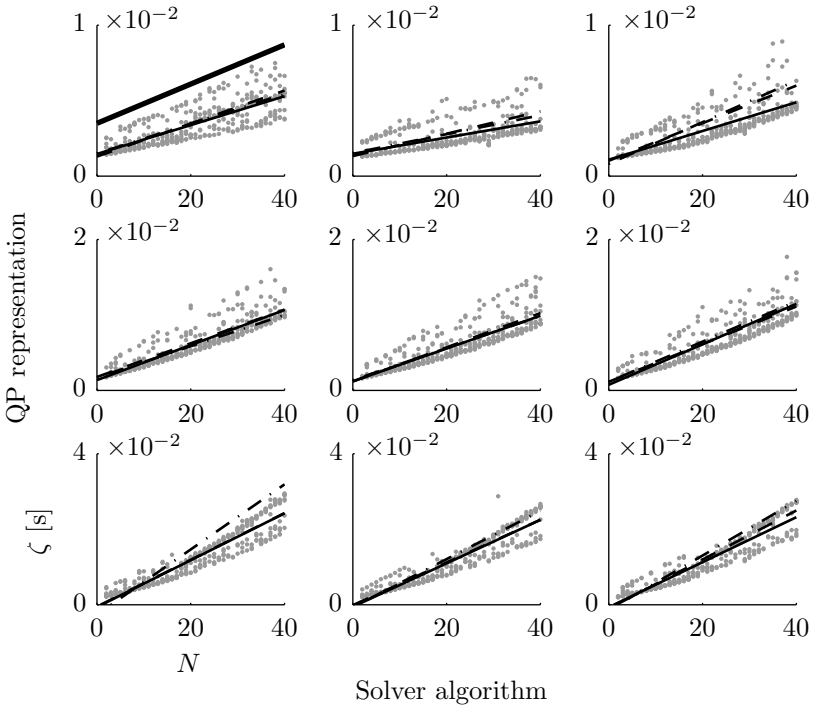

Figure 4.7: Relationship between solution time and number of prediction steps for different QP representations and QP solver algorithms. The upper-bound model ζ used in this paper is the thick solid line on the top-left graph. Left-to-right: Interior point, primal simplex and concurrent methods. Top-to-bottom: sparse, sparse-delta [LKC14] and dense QP representations. Data is obtained from the PAA problem (4.21) given in Section 4.5.1. Across all plots, grey point plots are data for $h = 5\,\text{ms}$ (point plots for other h values are not shown). The solid, dashed and dot-dashed thin black lines – mostly overlapping – are linear fits for the data with $h = 5\,\text{ms}$, 10 ms and 5 s.

function with the chosen constants $a_1 = 1.3 \times 10^{-4}$ and $a_0 = 3.5 \times 10^{-3}$ so that

$$\zeta(\mathbf{p}) = (0.13N + 3.5) \times 10^{-3} \tag{4.20}$$

The relevant data and model is shown on the top-left graph of Fig. 4.7.

4.5.1 Test plant models

The first test case looks at a missile pitch-axis autopilot (PAA) adapted from [BMM+14]. The missile is flown at a cruising altitude and the autopilot is to control the missile to track a commanded acceleration. The second case aims to design a controller for diesel engine control in submarines. The engine is modelled by a mean-value engine model (MVEM) taken from [BMBH15] and the control objective is to track a given engine speed and power output.

Pitch-axis autopilot

The first test case looks at a missile pitch-axis autopilot (PAA) at 20 000 ft. The study considers only the autopilot subsystem and serves as an introduction to the full missile control problem. The non-linear tracking model[4] is given as

$$\widetilde{\mathsf{f}}(\widetilde{\mathsf{x}}, \mathsf{u}) = \begin{bmatrix} \widetilde{\mathsf{x}}_2 + \cos(\widetilde{\mathsf{x}}_1) F_{\mathrm{L}}(\widetilde{\mathsf{x}}_1, \mathsf{u}) / (mv) \\ M(\widetilde{\mathsf{x}}_1, \mathsf{u}) / I_y \\ \widetilde{\mathsf{x}}_4 \\ -\omega_0^2 \widetilde{\mathsf{x}}_3 - 2\zeta\omega_0 \widetilde{\mathsf{x}}_4 + \omega_0^2 \widetilde{\mathsf{x}}_5 \\ \mathsf{u} \end{bmatrix} \tag{4.21}$$

where F_{L} and M are the non-linear mapping for the aerodynamic lift force and pitching moment respectively. $\widetilde{\mathsf{x}}_1$ is the angle of attack and $\widetilde{\mathsf{x}}_2$ is the pitch rate of the missile. The actuation of the fin deflection $\widetilde{\mathsf{x}}_3$ is modelled as a second order system. The input u is the rate of the commanded fin deflection $\widetilde{\mathsf{x}}_5$.

Missile speed $v = Mv_s$, where v_s is the speed of sound at 20 000 ft, is constant at Mach number $M = 2.5$. m and I_y are the mass and moment of inertia of the missile respectively. These parameter values, other missile frame parameters, constants related to the actuation dynamics, along with the aerodynamic coefficients and models used for F_{L} and L are the same as given in [BMM+14, NRR93].

[4] Symbols used in the plant model are used exclusively in the model and should not be confused with symbols introduced elsewhere.

The control objective is to track a given acceleration output $\mathsf{y} = F_\mathrm{L}/(mg)$ where g is the gravitational acceleration. The test scenario is to track five different acceleration outputs from steady state at $0g$, i.e. $\widetilde{\mathsf{x}}_0 = 0$. The outputs to track are 2, 4, 6, 8 and $10g$, each associated with a unique steady state $\widetilde{\mathsf{x}}_\mathrm{o}$ and initial (error) state as per (3.20). These scenarios make up the set of initial conditions $\mathcal{X}_0$ in (3.23) and are equally weighed, $w_i = 0.2$, $i \in \{0, \dots, 5\}$. The states are constrained with an upper-bound of $\overline{\widetilde{x}} = (20^\circ, 35^\circ/\mathrm{s}, 45^\circ, 10^{6\,\circ}/\mathrm{s}, 45^\circ)$ and lower bound of $\underline{\widetilde{x}} = -\overline{\widetilde{x}}$. These define the predictive state constraint according to (3.13). The input is upper- and lower-bounded by $\overline{u} = 10^{6\,\circ}/\mathrm{s}$ and $\underline{u} = -\overline{u}$ respectively. The stage cost weights for the states are set as

$$Q = C^\mathsf{T} C$$

where $C := \frac{d\mathsf{y}}{d\widetilde{\mathsf{x}}}\Big|_{\widetilde{\mathsf{x}}=\widetilde{\mathsf{x}}_\mathrm{o},\mathsf{u}=0}$ comes from the linearisation of the output y at the target steady state. For the input, $R = 10^{-6}$.

Diesel engine control with a mean-value engine model

The second case looks at engine control where the model is a 5-state 3-input mean-value engine model (MVEM) taken from [BMBH15]. The 5 states are the engine speed $\widetilde{\mathsf{x}}_1$, turbine speed $\widetilde{\mathsf{x}}_2$, VGT actuator position $\widetilde{\mathsf{x}}_3$, intake manifold pressure $\widetilde{\mathsf{x}}_4$ and temperature $\widetilde{\mathsf{x}}_5$. The three inputs are the injection duration u_1, load applied to the engine by the generator u_2 and the VGT commanded position u_3. The model[4] is given as

$$\widetilde{\mathsf{f}}(\widetilde{\mathsf{x}}, \mathsf{u}) = \begin{bmatrix} (\tau_\mathrm{eng} - \mathsf{u}_2)/J_\mathrm{e} \\ (P_\mathrm{t} - P_\mathrm{c})/(J_\mathrm{t}\widetilde{\mathsf{x}}_2) \\ (\mathsf{u}_3 - \widetilde{\mathsf{x}}_3)/\tau_\mathrm{VGT} \\ \dfrac{R_\mathrm{a}}{V_\mathrm{im}}(\dot{m}_\mathrm{c} + \dot{m}_\mathrm{EGR} - \dot{m}_\mathrm{ei})\widetilde{\mathsf{x}}_5 + \dfrac{\widetilde{\mathsf{x}}_4}{\widetilde{\mathsf{x}}_5}\dot{\widetilde{\mathsf{x}}}_5 \\ \dfrac{R_\mathrm{a}}{V_\mathrm{im}c_\mathrm{va}}\dfrac{\widetilde{\mathsf{x}}_5}{\widetilde{\mathsf{x}}_4}\Big(R_\mathrm{a}(T_\mathrm{ic}\dot{m}_\mathrm{c} + T_\mathrm{EGR}\dot{m}_\mathrm{EGR} - \widetilde{\mathsf{x}}_5\dot{m}_\mathrm{ei}) + \\ \qquad c_\mathrm{va}\dot{m}_\mathrm{c}(T_\mathrm{ic} - \widetilde{\mathsf{x}}_5) + c_\mathrm{va}\dot{m}_\mathrm{EGR}(T_\mathrm{EGR} - \widetilde{\mathsf{x}}_5)\Big) \end{bmatrix} \tag{4.22}$$

with static states determined after time-scale separation

$$0 = \begin{bmatrix} \dot{m}_{\mathrm{cyl}}(\mathsf{u}_1) - \dot{m}_{\mathrm{EGR}} - \dot{m}_{\mathrm{t}} \\ O_{\mathrm{cyl}}(\mathsf{u}_1) - O_{\mathrm{em}} \\ T_{\mathrm{em}} - \dfrac{c_{\mathrm{pe}}\dot{m}_{\mathrm{cyl}}T_{\mathrm{cyl}}(\mathsf{u}_1) + hA_{\mathrm{em,i}}T_{\mathrm{em,s}}}{c_{\mathrm{pe}}\dot{m}_{\mathrm{cyl}}(\mathsf{u}_1) + hA_{\mathrm{em,i}}} \end{bmatrix}$$

and the further assumptions that

$$0 = \begin{bmatrix} \widetilde{\mathsf{x}}_{\mathrm{EGR}} \\ O_{\mathrm{im}} - O_{\mathrm{FRs}} \end{bmatrix}.$$

J_{e}, J_{t}, τ_{VGT} and V_{im} represent physical engine parameters. R_{a} is the specific gas constant for the ambient gas. c_{va} is the isometric specific heat of the ambience and c_{pe} is the isobaric specific heat of the exhaust gas. O_{FRs} is the stoichiometric mass ratio. τ_{eng} is the engine load. P, $\dot{m}$ and T denote power outputs, mass flows and temperatures respectively. h and A denotes the heat transfer coefficient and its associated contact area. Subscripts *t*, *c*, *EGR* and *ic* represent associations with the turbine, compressor, EGR and intercooler respectively. *em*, *em,s* and *em,i* are associated with the exhaust manifold, whilst *cyl* and *ei* with the cylinders of the engine. Expressions for these algebraic variables are given in [BMBH15].

The initial state of the engine is at 2 000 rpm producing 20 kW of power. The control objective is for the engine to track a steady state at 2 500 rpm producing 36 kW of power. The states are upper-bounded by $\overline{\widetilde{x}} = (2500\,\mathrm{rpm}, 150\,000\,\mathrm{rpm}, 10^8, 10^8, 10^8)$ and lower-bounded by $\underline{\widetilde{x}} = (1500\,\mathrm{rpm}, 45\,000\,\mathrm{rpm}, 0, 0, 0)$. These define the predictive state constraint according to (3.13). The input is upper- and lower-bounded by $\overline{u} = (1\,\mathrm{ms}, 300\,\mathrm{Nm}, 90)$ and $\underline{u} = (0.5\,\mathrm{ms}, 10\,\mathrm{Nm}, 60)$. $Q = \mathrm{diag}(1,0,0,0,0)$ and $R = \mathrm{diag}(0,1,1)$ as to track engine speed, power output and VGT position.

4.5.2 Results

Fig. 4.8 shows a representative result for the PAA test case (4.21). A resulting trade-off curve is obtained after 20 evaluations using DITRI with $\overline{\mathbf{p}} = (0.015, 15)$ and $\underline{\mathbf{p}} = (0.001, 3)$, consisting of 10 different designs. For the MVEM test case (4.22), a representative result is shown on Fig. 4.9 for 20 evaluations using DITRI with $\overline{\mathbf{p}} = (0.4, 10)$ and $\underline{\mathbf{p}} = (0.05, 1)$. After 20 evaluations, 11 designs on a

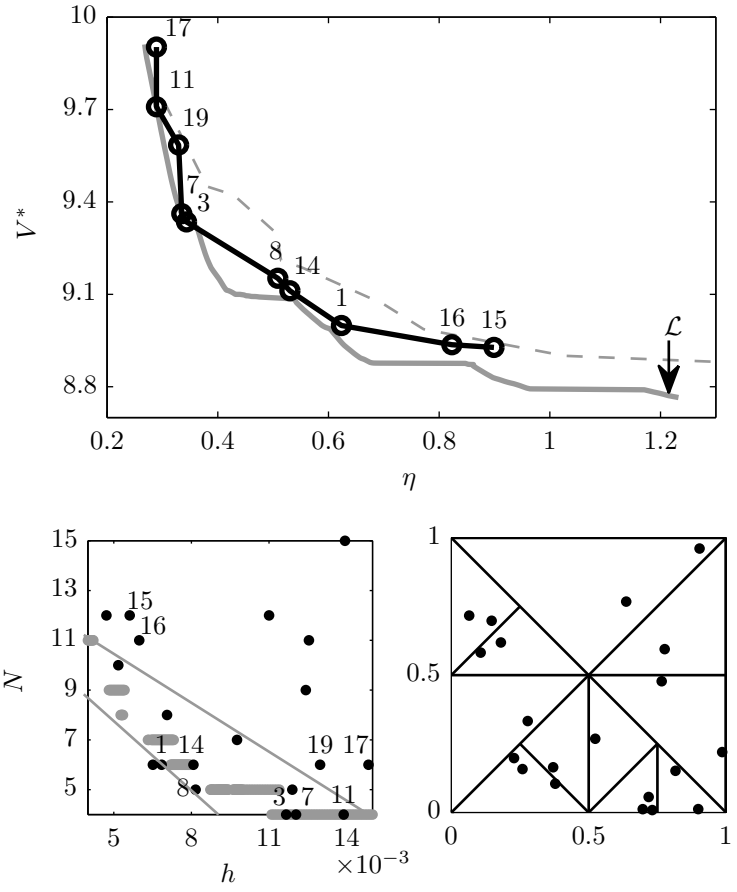

Figure 4.8: Pareto optimal solution for the design of the PAA controller[5].

trade-off curve are obtained. The associated solution in the design parameter space is shown on the bottom plots of each figure.

The trade-off curves represent the set of optimal designs a practitioner can choose from. For example, design 16 of the PAA test case (Fig. 4.8) has a controller design with a sampling time of $h = 6.5\,\text{ms}$ and $N = 11$ prediction steps. This gives a prediction horizon of around 70 ms. For this design choice, the implementation hardware of the controller should be able to solve the OCP with $N = 11$ in under 6.5 ms. To help design the implementation hardware, the capac-

[5]Top: trade-off curve (black) along with the Pareto front $\mathcal{L}$ obtained from a full design exploration (grey). Each point is labelled by the associated evaluation number. The dashed-grey line is a non-convergent HVOL solution. Bottom-left: the associated solution in parameter space along with the band of Theorem 4.5. Bottom-right: accompanying plot showing the normalised space c_1-c_2 and triangle divisions used internally in DITRI. Total number of evaluations is $\overline{i^{\text{ev}}} = 20$.

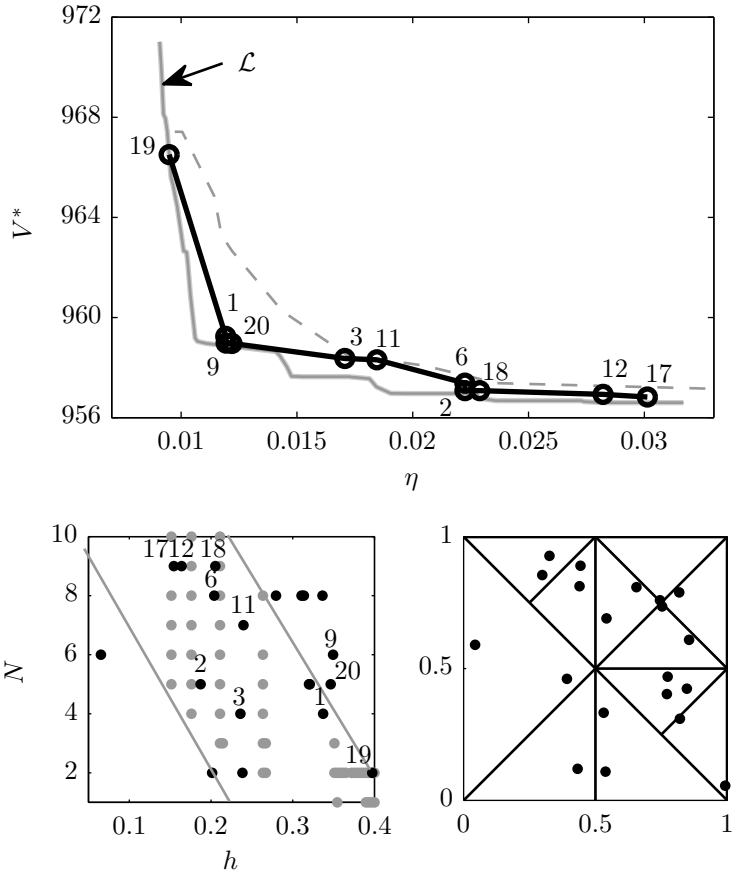

Figure 4.9: Pareto optimal solution for the design of the MVEM controller[5].

ity number of the design choice can be examined. PAA design 16 is associated with utilisation number $\eta = 0.82$, indicating that the implementation hardware must have at least around 0.8 times the processing power of the simulation hardware based on the specifications given in Table 4.1. This consideration depends on several factors of hardware capabilities and implementation architecture, including the use of compiled programming language, parallel-processing, an increased clock-frequency, as well as pipelining.

Finally, the trade-off curves reveals the sensitivity of control performance to computational capacity. In the PAA test case, performance improvement after utilisation number around $\eta = 0.6$ is not significant any more. This implies that there is not much benefit to be gained from hardware more powerful than around $\eta = 0.6$. In the MVEM test case, the value is around $\eta = 0.015$.

4.5.3 Validation of DITRI

To show the importance of the conditions for an effective and efficient optimiser prescribed in Section 4.4.2, DITRI is compared to two other algorithms. One is a non-dominated sorting genetic algorithm (NSGA) adapted from [DPAM02], and the other is an algorithm based on surrogate hyper-volume improvement (HVOL) adapted from [TSC13].

Table 4.2 outlines how each optimiser satisfies the conditions. As discussed in Section 4.4.2, DITRI is a optimiser which fully meets the prescribed conditions. NSGA handles integer variables directly and is an exhaustive search in the limit of infinite evaluation number, thus satisfies Condition 4.1. However, it is an all-purpose optimiser that does not rely on the continuity of the objectives. Similarly, HVOL is neither a continuous nor focussed search and so would not satisfy Conditions 4.2 and 4.3. Furthermore, although HVOL is able to handle integer variable and performs a global search, it is not guaranteed to converge [TSC13].

Effective convergence – Condition 4.1

To assess the convergence of the trade-off curves obtained, the curves are compared to the true Pareto front $\mathcal{L}$. Since the true Pareto front is not known, it is approximated by doing a full exploration on a uniform grid of 400 h values for each value of N in the parameter space. For both case studies, it is shown that the solution obtained by DITRI is close to the true Pareto front of the problem (Figs. 4.8 and 4.9).

The closeness of a trade-off curve to the true Pareto front can be measured by calculating the average of the closest Euclidean distance between each point on the trade-off curve to the Pareto front. This measure is denoted Ψ_{dis} and plotted in Fig. 4.10 against function evaluation count. A second metric calculates the Eu-

	Condition				
Algorithm	4.1	4.1a	4.1a	4.2	4.3
DITRI	✓	✓	✓	✓	✓
NSGA	✓	✓	✓		
HVOL		✓	✓		

Table 4.2: Fulfilment of conditions for convergence and efficiency.

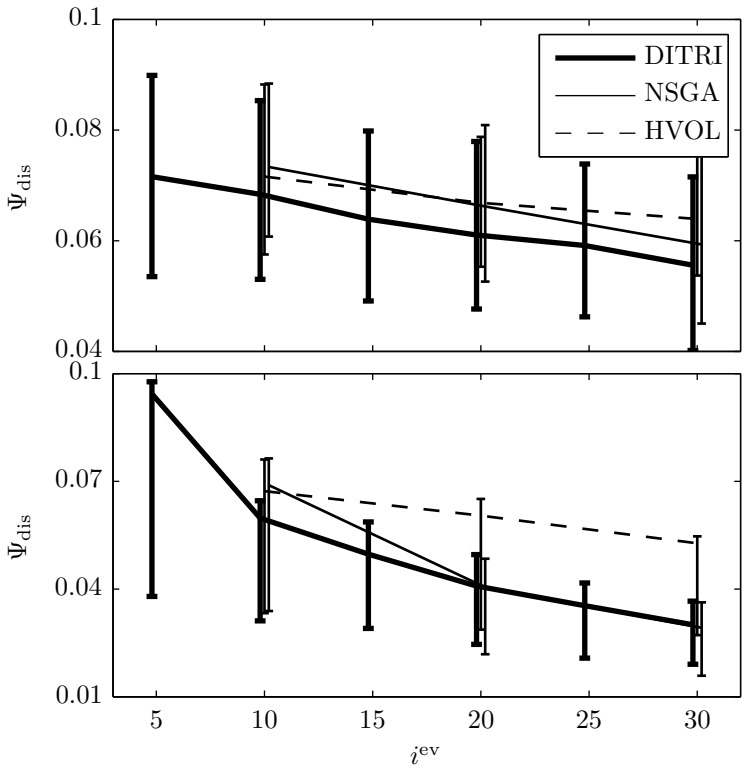

Figure 4.10: Closeness of solution against evaluation count for the PAA (top) and MVEM (bottom) test cases[6].

clidean distance of the tips (vertices) of the trade-off curve and Pareto front. This measures the coverage of the solution, denoted Ψ_{cov}, and is shown in Fig. 4.11. Calculation of both metrics are based on a normalised design objective values.

Figs. 4.10 and 4.11 show that trade-off curves produced by DITRI and NSGA approach the Pareto front with increasing function evaluation counts. However, HVOL struggles to converge. Figs. 4.8 and 4.9 show non-convergent trade-off curves, each from 100 evaluations using HVOL, confirming HVOL's inability to find the Pareto front. This is consistent with the expectation, since both DITRI and NSGA satisfies Condition 4.1 for convergence, whereas HVOL does not. The fulfilment of Conditions 4.1a and 4.1b by HVOL is not sufficient to guarantee convergence, consistent with Remark 4.5.

[6]The graphs show the mean across 500 trials with error bars showing the 25^{th} and 75^{th} percentiles.

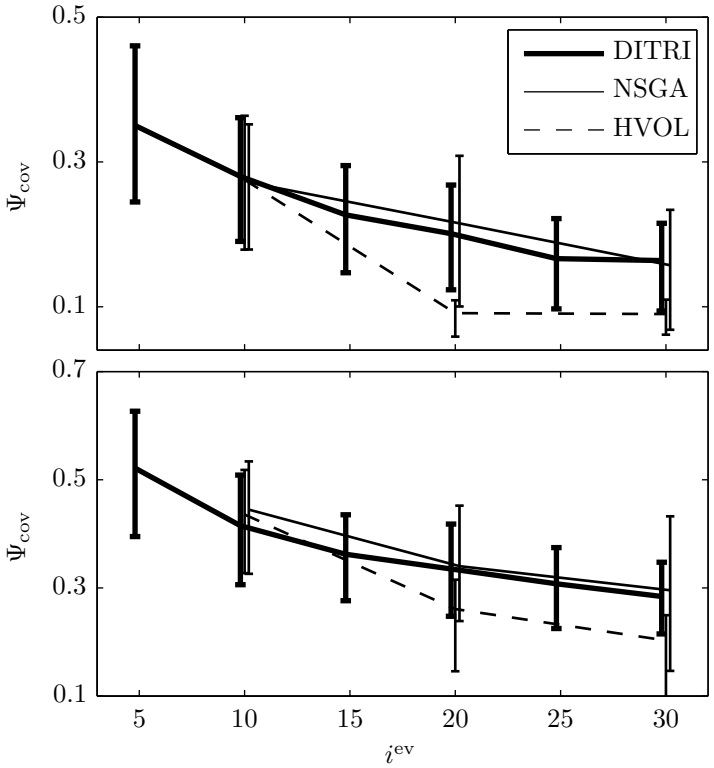

Figure 4.11: Coverage of solution against evaluation count for the PAA (top) and MVEM (bottom) test cases[6].

Efficient search – Conditions 4.2 and 4.3

The results in Figs. 4.8 and 4.9 show that the Pareto solution lies within a band as described in Theorem 4.5. DITRI takes advantage of this, unlike NSGA and HVOL. Comparing the results in Fig. 4.10, DITRI exhibits the best convergence rate to the overall Pareto front. This is consistent with the fact that DITRI satisfies Conditions 4.2 and 4.3 for efficiency, whereas the general-purpose NSGA and HVOL do not. In Fig. 4.11, results indicate that HVOL is better in terms of reaching to the tips of the Pareto front quickly. However, since HVOL does not guarantee overall convergence to the Pareto front solution (see Figs. 4.8–4.10), the fast convergence of HVOL should only be taken with a pinch of salt.

4.6 Conclusions

This chapter analytically studied the multi-objective design of MPC problem presented in Chapter 3 to present several theoretical results that reveal the nature of the design problem. Focus was given to tuning of structural parameters, namely the sampling time and prediction horizon length of the optimal control problem. The presented lemmas and theorems throughout the chapter outline key properties that include smoothness of the objective functions and bounds on the optimal design parameters. This is motivated by the need of an effective and efficient optimiser due to the time-consuming nature of the objective evaluations in the MOD-MPC problem. The findings that resulted from the analysis were subsequently used as a basis to prescribe necessary and sufficient conditions for such an optimiser. Subsequently, a compliant optimiser was proposed, developed based on Lipschitzian optimisation. Finally, two numerical tests on real-world examples were conducted to demonstrate the use of the proposed MOD-MPC design approach. The first test case investigated a missile autopilot problem in the pitch-axis as an introduction to the application of the thesis in missile control which will be considered in the next chapters. The numerical results also validated the sufficiency of the conditions specified for an effective and efficient optimiser of the design problem.

Chapter 5
Model-predictive integrated missile control

A substantial proportion of this chapter has been submitted for publication in the AIAA Journal of Guidance, Control, and Dynamics [BMMK16b].

EXIGENT mission requirements present a missile control problem that deals with non-linear, constrained and fast dynamics associated with the autopilot and guidance subsystems. Advanced missile control designs have been founded on recent developments that are aimed at a more streamlined engineering design approach. Central to these developments is the attempt to integrate the subsystems and the concept of co-design of multiple components in control system. In particular, the autopilot and guidance subsystems share an intrinsic relationship upon which a targeted integration of the two could utilise the specific synergies to improve the performance of the missile [MO01a]. In light of this, the chapter proposes a model-predictive integrated missile autopilot and guidance (iMAG). The development of the integrated controller follows the application of MPC for the pitch-axis autopilot plant model in Chapter 4 to present the full missile control system including both autopilot and guidance.

5.1 Engagement dynamics

Consider first the equations of motion for an engagement between a missile and an incoming target. The model can be separated into its dynamics and kinematics part. A cruciform, tail-controlled, and roll-stabilised missile is considered in this study. Focus is given on the end-game phase of the flight, in which the target can be assumed to be relatively close, and control can be separated into two equivalent planar channels that are perpendicular to each other. Modelling of the missile equations of motion is presented for control in one channel for pitch control. Note that since a cruciform missile is considered, the pitch control study can be applied to that of yaw due to the symmetry of the missile.

5.1.1 Missile dynamics

The missile dynamics are defined around the body and wind axes. The body axis is fixed to the physical frame of the missile, whereas the wind axis is aligned to the velocity of the missile (see Fig. 5.1). Both axes are centred at the center of gravity of the missile. The missile translational and rotational dynamics are described with respect to the body axis, defined as follows

$$\dot{\alpha} = q - F_{\mathrm{L}}(\alpha, \delta)/(mV) \tag{5.1}$$

$$\dot{\theta} = q \tag{5.2}$$

$$\dot{q} = M(\alpha, \delta)/I_y \tag{5.3}$$

α is the angle of attack and q is the pitch rate of the missile. The missile speed $V_{\mathrm{M}} = M_a V_{\mathrm{s}}$, where V_{s} is the speed of sound, is constant at Mach number $M_a = 2.5$. The missile is flying at speed and altitude with dynamic pressure P. The actuation of the effective control surface deflection δ for pitching is modelled as a second order system,

$$\dot{\delta} = \dot{\delta} \tag{5.4}$$

$$\ddot{\delta} = \omega_a^2(\delta_{\mathrm{c}} - \delta - 2\xi\dot{\delta}) \tag{5.5}$$

$$\dot{\delta_{\mathrm{c}}} = \mathsf{u} \tag{5.6}$$

where the commanded fin deflection δ_{c} rate is the input u of the system. Aerodynamic lift F_{L} and moment M are modelled to be cubic with respect to the angle of attack and linear with respect to the control surface deflection [BMM+14,NRR93].

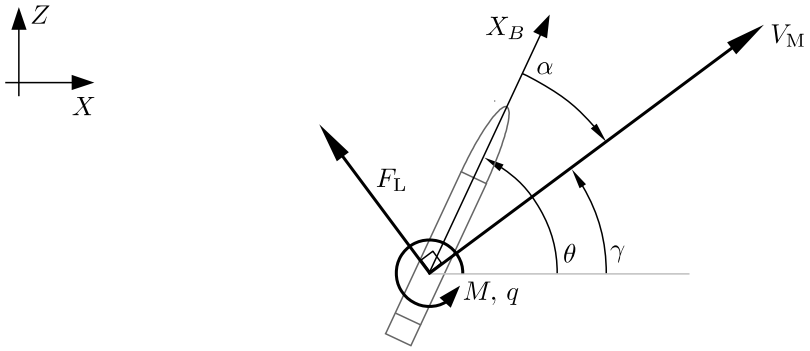

Figure 5.1: Frame for the dynamics of the missile.

That is,

$$F_{\mathrm{L}} = PSC^L \qquad\qquad M = PSDC^M \tag{5.7}$$

where the aerodynamic coefficients are

$$C^L := \sum_{i=1}^{3} c_i^F \alpha^i + c_\delta^F \delta \qquad\qquad C^M := \sum_{i=1}^{3} c_i^M \alpha^i + c_\delta^M \delta. \tag{5.8}$$

c is the stability and control derivatives of the missile, describing how forces and moments (denoted by its superscript) change with respect to its parameters (denoted by its subscript).

5.1.2 Engagement kinematics

The kinematics of the missile is defined with respect to a fixed inertial coordinates, e.g. ground/earth-based, as illustrated in Fig. 5.2. The missile flight-path angle is denoted by γ. The seeker detects the location of the target relative to the missile, which is defined at a distance r and angle χ. Based on the seeker and pitch angles, χ and θ, a line-of-sight (LOS) angle λ can be derived, which measures the relative location of the target to the missile with respect to the fixed inertial axis.

The evolution of the missile location in Cartesian coordinates $\mathbf{s}_{\mathrm{M}} := (s_{\mathrm{M1}}, s_{\mathrm{M2}})$ with respect to the inertial axis is given by

$$\dot{s}_{\mathrm{M1}} := V_{\mathrm{M}} \cos\gamma \qquad\qquad \dot{s}_{\mathrm{M2}} := V_{\mathrm{M}} \sin\gamma$$

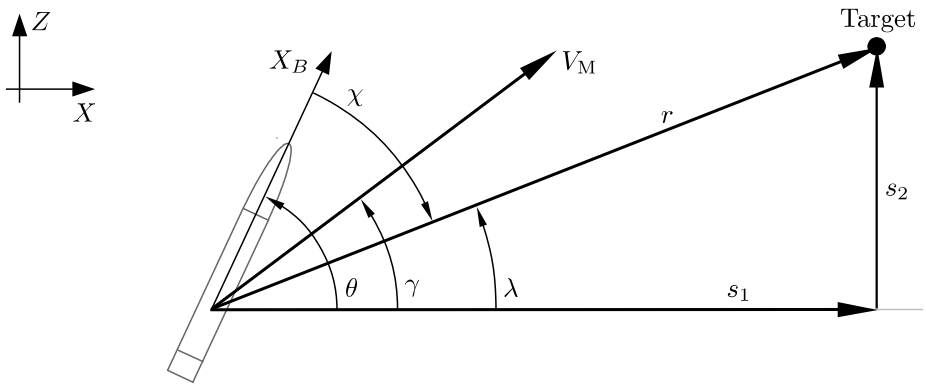

Figure 5.2: Frame for the kinematics of the missile and target.

where $\gamma := \theta - \alpha$. The target is assumed to have a constant acceleration $\mathbf{a}_\mathrm{T}$. The target location and velocity is denoted by $\mathbf{s}_\mathrm{T}$ and $\mathbf{v}_\mathrm{T}$,

$$\dot{\mathbf{s}}_\mathrm{T} =: \mathbf{v}_\mathrm{T} \qquad\qquad \dot{\mathbf{v}}_\mathrm{T} =: \mathbf{a}_\mathrm{T}.$$

In relative terms, the separation between the missile and target is denoted by $\mathbf{s}$ and modelled as

$$\dot{s}_1 =: v_1 = v_{\mathrm{T}1} - V_\mathrm{M}\cos\gamma \qquad\qquad \dot{s}_2 =: v_2 = v_{\mathrm{T}2} - V_\mathrm{M}\sin\gamma \tag{5.9}$$

and the relative acceleration between the missile and the target is

$$\dot{v}_1 = a_{\mathrm{T}1} - \frac{F_\mathrm{L}(\alpha,\delta)}{m}\sin(\theta-\alpha) \qquad \dot{v}_2 = a_{\mathrm{T}2} - \frac{F_\mathrm{L}(\alpha,\delta)}{m}\cos(\theta-\alpha). \tag{5.10}$$

The LOS angle and its evolution in time can be derived from the relative distances $\lambda = \arctan(s_2/s_1)$ such that the LOS second order rate is given as

$$\ddot{\lambda} = \frac{\left(s_1^2+s_2^2\right)\left(s_1\ddot{s}_2 - s_2\ddot{s}_1\right) + 2\left(\dot{s}_1 s_2 - s_1\dot{s}_2\right)\left(s_1\dot{s}_1 + s_2\dot{s}_2\right)}{\left(s_1^2+s_2^2\right)^2}. \tag{5.11}$$

5.1.3 Engagement model

Summarizing the kinematics and dynamics is a plant model with states $\mathsf{x} := (\alpha, \theta, q, \delta, \dot{\delta}, \delta_c, \mathbf{s}, \mathbf{v}, \dot{\lambda})$ and control input $\mathsf{u} := \dot{\delta}_c$. Put concisely,

$$\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x},\mathsf{u}). \tag{5.12}$$

where f is obtained from (5.1)–(5.6) and (5.9)–(5.11).

5.2 Model-predictive integrated missile controller formulation

In this section, the control system for the missile is formulated. As detailed in Chapter 2, an integrated system has been the focus in advanced missile control technology and will be used as a basis for the following controller design development. A model-predictive control approach provides the means to handle the non-linear and constrained dynamics of the engagement.

5.2.1 Control strategy

In this study, a control strategy for intercept will be employed. Intercept is achieved when the relative position of the missile and the target $\mathbf{s}$ is zero. This is contrasted to, for example, a rendezvous strategy where in addition to a zero

relative distance, the lateral relative velocity between the missile and the target is required to be zero.

Remark 5.1. Although intercept is ideally achieved by an on-target hit, non-zero miss-distances can still result in a successful engagement as anti-missiles in missile defence typically carry a warhead which allows for the missile to incapacitate the target for a non-zero miss-distance [Zar99].

The proposed model-predictive integrated control scheme is based on a *parallel navigation* that regulates the rate of the LOS angle λ that dictates the relative trajectory between the missile and the target. Parallel navigation relies on the fact that a zero LOS rate during engagement is a sufficient condition for the missile to intercept the target given that the missile is approaching the target (see, for example, [PBL10b]). Therefore, the control strategy employed will be to steer the missile such that the LOS rate $\dot{\lambda}$, whose evolution is as described in (5.11), is regulated to zero.

As the relative distance approaches intercept at $\mathbf{s} = 0$, the LOS second order rate given in (5.11) approaches infinity,

$$\begin{aligned} &\text{if} \quad \dot{\mathbf{s}} \neq 0 \text{ or } \ddot{\mathbf{s}} \neq 0 \text{ or } \dot{s}_1 \neq \dot{s}_2 \text{ or } \ddot{s}_1 \neq \ddot{s}_2 \\ &\text{then} \quad \lim_{s1 \to 0,\, s2 \to 0} \ddot{\lambda} = \infty. \end{aligned} \tag{5.13}$$

As the missile approaches the target, relative velocity and acceleration is non-zero throughout the engagement and its components are not guaranteed to be equal. Therefore, the conditions in (5.13) are satisfied and thus it is inevitable that the LOS and its rates deviates significantly at the end of the engagement.

Remark 5.2. Note that the LOS can also be derived from the measurable angles $\lambda = \theta - \chi$ such that

$$\dot{\lambda} = q - \dot{\chi} \tag{5.14a}$$

$$\ddot{\lambda} = \dot{q} - \ddot{\chi}. \tag{5.14b}$$

In simulation, the angles q, χ, and their derivatives have to be calculated based on the relative distances, so there is no advantage in using (5.14). However, in practice, the seeker angle and its rate are readily available from the inertial measurement unit (IMU) and seeker of the missile, whereas the distances $\mathbf{s}$ might be more difficult to obtain directly. Therefore, (5.14) might be a preferable alternative to (5.11) in the actual implementation of the controller.

5.2.2 OCP formulation with measurement disturbance

The missile is controlled by commanding the fin deflection rate $\mathsf{u} \coloneqq \dot{\delta}_c$ to steer the missile. Control is commanded in a sampled-data fashion at sampling instants t_i, $i \in \mathbb{N}_0$, with sampling interval h, and is restricted to a zero-order-hold (ZOH). Let the control law be denoted by

$$\mathsf{u}_i = \kappa(\mathsf{o}_i, \mathbf{p})$$

to follow (3.1). The control is calculated based on the measured states o_i available at sampling instant t_i, as well as control design parameters $\mathbf{p}$. The control states o_i are measured from the real-time states x_i that are relevant to the prediction model, obtained via sensors. To better represent real-world conditions, a non-zero disturbance from sensor imperfections is allowed, and will be modelled by

$$\mathsf{o}_i \coloneqq \mathsf{x}_i + w. \tag{5.15}$$

with $w \in \mathcal{W} \subset \mathbb{R}^{n_\mathsf{x}}$ for some closed set $\mathcal{W}$.

A controller based on model-predictive control (MPC) is formulated following the outlined dynamics and kinematics of the missile. The control law is obtained by solving an optimal control problem (OCP),

$$(\mathbf{x}^*, \mathbf{u}^*) \coloneqq \arg\min_{(\mathbf{x},\mathbf{u})} \overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{5.16a}$$

$$\text{s.t. } x_0 = \mathsf{o}_i \coloneqq \mathsf{x}_i + w \tag{5.16b}$$

$$x_{k+1} = \overline{f}(x_k, u_k, \mathbf{p}) \coloneqq x_k + \int_{t_i}^{t_i+h} f(x(\tau), u_k, \mathbf{p})\, d\tau \qquad \forall k \in \{0, \dots, N_\mathrm{e} - 1\} \tag{5.16c}$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U} \qquad \forall k \in \{0, \dots, N_\mathrm{e} - 1\}. \tag{5.16d}$$

Consequently, the control law is given by

$$\mathsf{u}(t) = \mathsf{u}_i = \kappa(\mathsf{o}_i, \mathbf{p}) \coloneqq u_0^* \qquad \forall t \in [ih, ih + h), i \in \mathbb{N}_{\geq 0}. \tag{5.17}$$

The proposed control law is based on the discrete-time non-linear MPC (NMPC) introduced in (3.9) with predictive states and inputs as per (3.8), constituting the model-predictive integrated autopilot and guidance proposed in this study. The use of a non-linear prediction model, as opposed to a linear one used in previous chapters, is to better capture the full dynamics of the prediction model of the missile engagement.

The OCP optimisation is subject to the (discrete-time) prediction model (5.16c) representing the dynamics of the plant. This is based on the continuous-time counterpart

$$\dot{x} = f(x, u)$$

with the states $x := (\alpha, \theta, q, \delta, \dot{\delta}, \delta_c, \mathbf{s}, \mathbf{v}, \dot{\lambda})$ and input $u := \dot{\delta}_c$. The first six prediction model states are the missile dynamics modelled as (5.1)–(5.6). The last three states in the prediction model are required for the modelling of the LOS angle λ (5.9)–(5.11). The inclusion of the LOS angle in the prediction model used by the controller is required by the control strategy based on the regulation of LOS rate. Note the distinction to the real-time plant model $\dot{\mathsf{x}} = \mathsf{f}(\mathsf{x}, \mathsf{u})$. The distinction allows for the fact that the predicted dynamics does not necessarily always match the actual evolution of the states. The model is initialised at o_i as given in (5.16b), where o_i consists of the relevant states available at sampling instant t_i from possibly imperfect measurements. Finally, note that since the control strategy is the regulation of LOS rate, the regulation-tracking transform (3.3) is not needed.

There is an operating envelope within which aerodynamic models used are valid and the missile is able to operate stably. A constraint on the angle of attack α is required to keep the flight conditions within an envelope where aerodynamic forces can be accurately represented by the models (5.8). Fin deflection δ and its rate $\dot{\delta}$ are mechanically limited by the actuator. Finally, the angle χ may be constrained to naturally reflect the effective scope of the seeker. These are modelled by the sets $\mathcal{X}$ and $\mathcal{U}$ in the constraints (5.16d) for which the prediction model states and inputs are allowed to be in.

The cost function (5.16a) is chosen as a quadratic as per (3.10),

$$\overline{J}(\mathbf{x}, \mathbf{u}, \mathbf{p}) := \sum_{k=0}^{N_e-1} \left\| \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^T & \overline{R} \end{bmatrix}} + \|x_{N_e}\|^2_{Q_t}$$

The prediction horizon length is set to be an integer multiple of the sampling time, $T := N_e h$ associated with N_e prediction steps. The number of prediction steps is chosen to be

$$N_e := \min\left\{N, \lfloor t_{go}/h \rfloor\right\} \tag{5.18}$$

where the time-to-go $t_{\text{go}} := \|\mathbf{s}\|/\|\mathbf{v}\|$, to ensure that the prediction does not exceed the point beyond intercept. This is motivated by the fact that the LOS second order rate $\ddot{\lambda}$ is very sensitive around the intercept as shown in (5.13), and would easily cause numerical errors.

A natural choice for the weights is to penalise first and foremost the LOS rate. Secondly, the pitch rate will also be penalised, with relatively less weight than LOS rate, to avoid chattering. The terminal cost weight Q_{t} will be set in a similar fashion to the state stage cost weight for the same reasons. For the input, the fin deflection rate, a small non-zero is desired to avoid chattering and ensuring a positive definite cost function and uniqueness of the OCP solution.

Remark 5.3. The strategy of LOS rate regulation is employed as a zero LOS rate throughout the engagement is feasible to be achieved. This is different to a separation-nullifying strategy, e.g. in [KKK00], as the distance between the missile and the target can only be zero at the end of the engagement. In an MPC setting, when a shorter fixed prediction horizon (often shorter than the time-to-go) is desirably used to reduce computational cost, the later strategy is susceptible to undesirable control such as large overshoots.

5.3 Simulation illustrations

5.3.1 Preliminaries

Missile specifications

The missile used in this study is a tail-controlled axisymmetric ogive-nosed missile that is flying at a cruising altitude of 20 000 ft. Fig. 5.3 shows the shape of the missile along with the dimensions proportional to the diameter D of the missile. The physical, actuation and flight specifications of the missile is given in Table 5.1.

The models for aerodynamic lift and moment (5.7) of the missile are obtained from DATCOM [VSB$^+$88], yielding the polynomial models

$$C^L = 9.0126\alpha + 46.4585\alpha^3 + 3.2390\delta$$
$$C^M = -20.1395\alpha - 51.7438\alpha^3 + 843.4588\alpha^5 - 24.1742\delta.$$

Symbol	Description	Value	Unit
D	diameter	0.2286	m
S	surface area	0.04088	m^2
m	mass	204	kg
I	moment of inertia	247.4	kg m^2
ξ	Actuator damping ratio	0.7	-
ω_a	Undamped natural frequency	150	rad/s
V_s	speed of sound	315.9	m/s
V_M	missile speed	$2.5V_s$	m/s
P	dynamic pressure	210 000	Pa

Table 5.1: Physical, actuation and flight specifications of the missile. Adapted from [BMM+14, NRR93].

Initial conditions, constraints and disturbance

The engagement scenarios considered are illustrative of a typical missile defence where the target is a hypersonic cruise missile. Note that the specified scenarios, which are adopted from [PBL10b], are particularly challenging, chosen to represent the high requirements in advanced missile control. The initial conditions represent the final stages of the engagement where the missile control is employed, after the boosting and cruising phases. The missile is initially positioned at the origin, travelling in the crossrange at Mach 2.5. Two target scenarios are considered. For the first scenario, the target is positioned at 4000 m crossrange and 0 m downrange, and for the second, the target location is 5000 m and 300 m cross and downrange, respectively. For both cases, the target is travelling at Mach 5, representing the speed of a typical cruise missile, in the crossrange to-

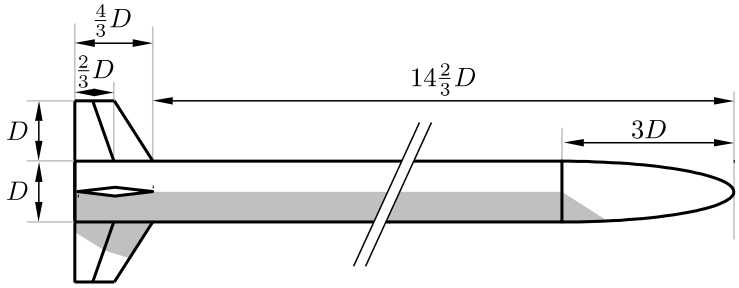

Figure 5.3: Missile dimensions. Adapted from [BMM+14, NRR93].

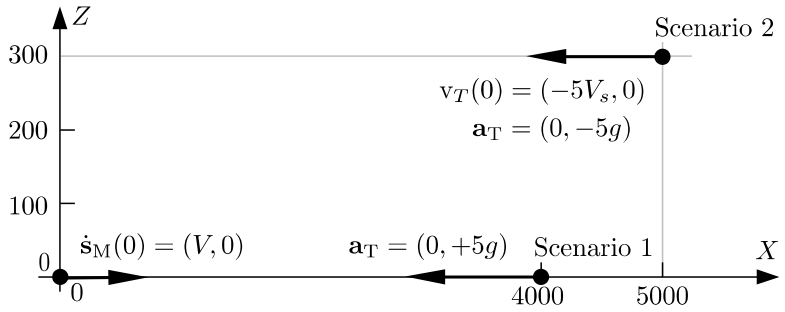

Figure 5.4: Initial missile and target positions, speeds and accelerations.

wards the missile. At the given speeds, the missile has to accelerate and adjust its direction in about only 2 seconds before it passes the target. The acceleration of the target is at $5g$ either in the positive or negative downrange for the respective scenarios. Finally, the initial missile location and all dynamic states are initialised at zero. This is illustrated in Fig. 5.4.

The constraint sets $\mathcal{X}$ and $\mathcal{U}$ that bounds the predictive states and inputs are chosen as polytopes formed by upper- and lower-bounding each state and input; $\mathcal{X} = \{x \,||x| \leq \overline{x}\}$ and $\mathcal{U} = \{u \,||u| \leq \overline{u}\}$, where the bound for the states and input are

$$\overline{x} = (20^\circ, 90^\circ, 500^\circ/\text{s}, 40^\circ, 500^\circ/\text{s}, 40^\circ, \infty, \infty, \infty, \infty, \infty)$$
$$\overline{u} = 500^\circ/\text{s}.$$

The disturbance w in (5.15) is modelled to arise from imperfections of the LOS angle λ sensor readings. That is,

$$w_{\dot{\lambda}} = \mathsf{U}[-0.02, 0.02]. \tag{5.19}$$

Where the subscript $\dot{\lambda}$ indicates that the disturbance acts only on the LOS rate. The disturbance is time-varying and changes for each sampling instant (regardless of the sampling time length).

OCP formulation and solver

In the illustrative simulations that follow, sampling time is set at $h = 15\,\text{ms}$ and the horizon length at $N = 10$ prediction steps. The cost weights are chosen as

$$\overline{Q} = \overline{Q}_\text{t} = \text{diag}\left(\mathbf{0}_{10\times10}, 5 \times 10^3\right) \qquad \overline{S} = \mathbf{0}_{11\times1} \qquad \overline{R} = 1 \times 10^{-6}.$$

For the results, simulations are done in MATLAB on a desktop computer with specifications as outlined in Table 4.1. The non-linear OCP (5.16) is solved using a sequential quadratic program (SQP) algorithm from [BMM+14] that is outlined in Appendix B.1. The interior point (barrier) method from the Gurobi solver [Gur13] in MATLAB is used in this study to solve each QP iteration in the SQP algorithm.

5.3.2 Simulation results

A simulation is carried out to demonstrate how the proposed control strategy guides the missile to engage a target. First, a case with zero disturbance is considered. Figs. 5.5 and 5.6 show the evolution of the dynamic and kinematic states of the missile for the two engagement scenarios. For the first scenario, within the first half second or so, the system is mostly transient with angle-of-attack α lying close to the constraint of 20°. The fin deflection δ is quite actively controlled as the missile accelerates upwards (positive $\ddot{s}_{\mathrm{M2}}$) to regulate LOS rate $\dot{\lambda}$. After around 0.5 s, the missile reaches close to a steady state as it moves to intercepts the target. The same behaviour is also observed in the first second of the second engagement scenario.

The proposed control strategy has a prediction horizon (5.18) that is shortened at the final stages of the engagement so that the prediction does not exceed the intercept point. As a result, at the point of shortening, in this case around 0.15 s before intercept, there is a slight transient as the prediction horizon length is shortened to not predict beyond the instant of intercept. Intercept happens at around $t = 1.7$ s and 2.1 s for the first and second scenario respectively, as shown by the path of the missile and target. A large deviation in LOS rate $\dot{\lambda}$ is observed at the final stages of the engagement. This is due to the fact that as the missile and target gets closer, the LOS second order rate $\ddot{\lambda}$ deviates significantly as discussed earlier and shown in (5.13).

For the considered simulations, it is observed that the missile intercepts the target with near zero miss-distance under no disturbance. When subject to disturbance, the miss-distance value is distributed over a set of simulations. Fig. 5.7 plots the histogram of miss-distance values for 100 simulations for the disturbance formulated in (5.19). Note that a non-zero miss-distance may be regarded as a successful intercept to some allowable degree as per Remark 5.1. The figure shows that the proposed missile controller achieves miss-distances much closer

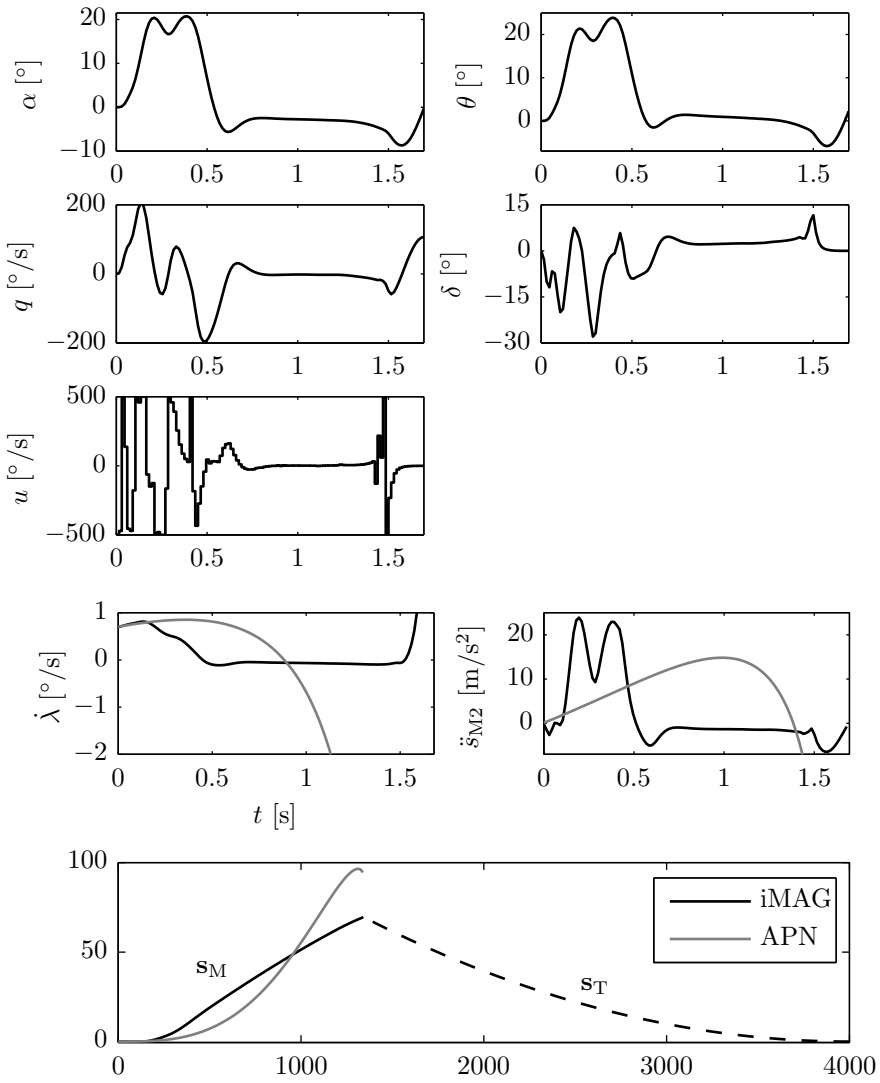

Figure 5.5: Evolution of dynamic and kinematic states for engagement scenario 1. $h = 0.015$, $N = 10$.

to zero than that of APN. With smaller miss-distances, the missile would be more effective in intercepting the target. For instance, the detonation of the warhead will be more devastating given the close proximity of the target, or lighter warhead explosive payloads can be used that allows for a lighter, more agile missile.

The main advantages of the model-predictive integrated missile control, i.e. iMAG, over a separated guidance-autopilot system are demonstrated by the LOS rate and acceleration plots in the second bottom row of Figs. 5.5 and 5.6, as well

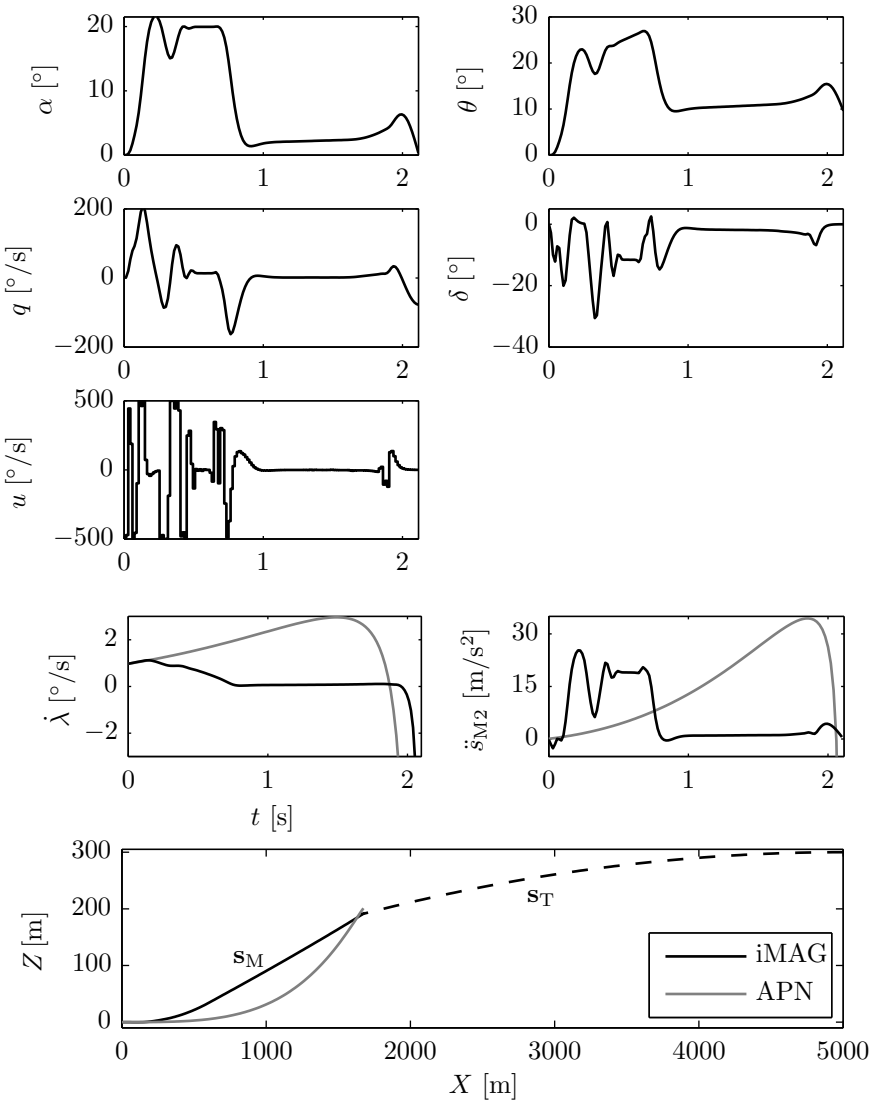

Figure 5.6: Evolution of dynamic and kinematic states for engagement scenario 2. $h = 0.015$, $N = 10$.

as Fig. 5.7. The plots are accompanied by the result obtained using an augmented proportional navigation guidance (APN) with an assumed ideal autopilot which can track commanded accelerations perfectly (see Appendix A.2). These superiority of iMAG to APN are given in the following.

1. *Look ahead*. Using the prediction from the model-based integrated controller, the missile can correct its path as soon as the engagement starts and undergoes large accelerations. During the second half of the engagement period, the mis-

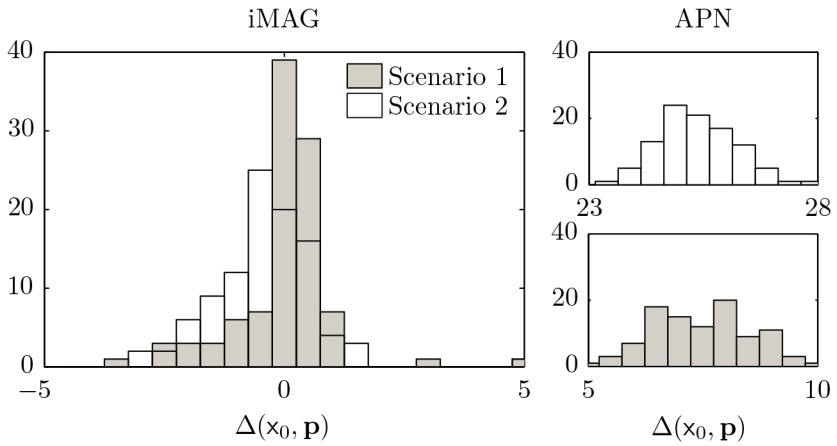

Figure 5.7: Histogram of miss-distances over a set of 100 simulations. The results for iMAG are shown on the left and APN on the right. For the former, results for both engagement scenarios are overlaid, whereas results are separated for the latter.

sile is mainly in constant velocity with minimal acceleration. At this stage, the missile is in a 'neutral' state and ready for any change in target acceleration obtained from the seeker. In conventional separated guidance-autopilot systems (e.g. with APN guidance) the missile has a non-zero acceleration throughout the engagement period, even under ideal conditions of no autopilot lag, so that it is at no time fully ready for changes in target manoeuvre. This is consistent with results in [PBL10b].

2. *Zero acceleration lag*. In convectional guidance systems (e.g. APN), discrepancies between the guidance command and the lagged actual acceleration achieved by the autopilot in separated systems can cause an undesirable growth in commanded acceleration as the instant of intercept is approached [SDBP14]. In these systems, commanded acceleration is directly proportional to LOS rate which grows unbounded close to intercept as formalised in (5.13). Consequently, the presence of acceleration lag in tracking would cause a deterioration in missile accuracy. Fig. 5.7 shows that APN, even with ideal autopilot, generally fails to achieve zero miss-distance.

5.4 Conclusions

A model-predictive integrated missile control was developed, aimed to improve control performance by commanding optimal missile accelerations. Through numerical simulations, it was shown that the proposed model-predictive integrated control system pushes the missile to be more responsive than when a conventional separated guidance-autopilot system is used. This was achieved by the high-performance nature of the proposed MPC-based controller that predicts future kinematics and dynamics of the engagement and finds the optimal missile accelerations under the operational constraints. Furthermore, the integrated design circumvents control lag present when the guidance-autopilot subsystems are implemented separately, improving accuracy under disturbance and ultimately the missile effectiveness in intercepting the target.

Chapter 6
Multi-objective design in advanced missile control

A substantial proportion of this chapter has been submitted for publication in the AIAA Journal of Guidance, Control, and Dynamics [BMMK16b].

THIS chapter investigates a case study following analytical results that has been developed so far in previous chapters. In Chapter 5, a model-predictive integrated missile control is proposed, extending the pitch-axis autopilot case considered in Chapter 4. In this chapter, the multi-objective design approach formulated in Chapter 3 is demonstrated on the proposed integrated missile control. In the earlier demonstrations of Chapter 4, control performance is represented by the closed-loop value function – measuring the integrated squared error of the states and inputs – whose smoothness properties with respect to the structural design parameters sampling time and prediction horizon are specified for when a linear prediction model is used. In this chapter, the investigation is extended to incorporate a more specific measure for performance that is based on the miss-distance of the missile when it passes the target. Such a measure has a better practical meaning in capturing control performance, and can be assumed to retain its competitive pairing with required computational capacity. Moreover, the MPC-based integrated missile control uses a non-linear prediction model for the OCP. The demonstration of the MOD-MPC approach on an application with non-linear MPC and non-linear plant numerically validates the theories in Chapter 4 to complement and complete the investigation of a linear prediction model in the earlier chapters.

6.1 Miss-distance as a performance metric

One underlying design objective in a missile control system is the performance of the controller based on the ability of the missile to intercept the target and how well the controller guides the missile in doing so, especially under disturbance. In missile engagement, performance can be generally measured based on the miss-distance, that is how close the missile intercepts the target [Ohl96].

Such a measure is specific for a missile engagement and can be interpreted more easily than the more MPC-general closed-loop value function (3.21) whose units are rather arbitrary. In addition to the miss-distance, the evolution of the states can also be qualitatively observed to see how the controller guides the missile in intercepting the target.

An engagement scenario can be numerically simulated by integrating the model (5.12) controlled by the MPC law (5.17) with the OCP (5.16) under disturbance in the measured states (5.15). A set of simulations will be performed to obtain the distribution of the performance. For a simulation, the miss-distance as a metric for control performance can be calculated based on the relative trajectory $\mathbf{s}$ of the missile and the target for a time period T_{sim} given an initial state x_0,

$$\begin{aligned} \Delta(\mathsf{x}_0, \mathbf{p}) &:= s_{\text{sgn}} \left(\min_{\tau \in [0, T_{\text{sim}}]} \|\mathbf{s}(\tau)\| \right) && \text{for } \tau \in [0, T_{\text{sim}}] \\ &\text{s.t. dynamics (5.12) and control law (5.17)} && \text{for a given } \mathsf{x}(0) = \mathsf{x}_0. \end{aligned} \tag{6.1}$$

The simulation time T_{sim} should be large enough to allow for the missile to reach the target. An indicative lower-bound of this time would be $\|\mathbf{s}(0)\| / \|\mathbf{v}(0)\|$. The multiplier $s_{\text{sgn}} := \text{sgn}\left(s_{M_2}(T_{\text{s}}) - s_{T_2}(T_{\text{s}})\right)$ indicates the relative position of the missile when it reaches the target, differentiating between the missile passing 'above' or 'below' the target. The average value of the distributed miss-distance is defined as

$$\overline{\Delta}(\mathsf{x}_0, \mathbf{p}) := \text{mean}(|\Delta(\mathsf{x}_0, \mathbf{p})|)$$

with a standard deviation denoted as $\sigma_\Delta(\mathsf{x}_0, \mathbf{p})$.

6.2 Solution time of SQP for required computational capacity

Non-linear MPC (NMPC) is used in the proposed model-predictive integrated missile control as defined in (5.16). The associated non-linear OCP is solved with the sequential quadratic programming (SQP) algorithm as outlined in Appendix B.1. The solution time associated with the SQP solver will be denoted ζ_{SQP}.

From Remark 3.3 and Assumption 3.6, the solution time of a quadratic programming (QP) optimisation can be upper-bounded by a polynomial of N,

denoted by ζ (3.26), which dictates the size of the QP, given that the cost function is positive definite as per Remark 3.1 [KTK80]. In this study, the non-linear OCP (5.16) is solved using a sequential quadratic program (SQP) algorithm [BMM+14]. The routine performs a number of QP optimisation to iteratively approach the non-linear OCP solution. The number of QP iterations the SQP algorithm performs can be upper-bounded by a specified iteration limit $\bar{i}_{\text{SQP}}$. Therefore, the solution time ζ_{SQP} associated with the problem can be upper-bounded by a polynomial of N.

Proposition 6.1 (SQP solution time upper-bound)**.** *From Assumptions 3.6 and 3.7, the solution time of an OCP as a quadratic program can be upper-bounded by a polynomial of degree n with respect to N,*

$$\zeta(\mathbf{p}) \coloneqq \sum_{i=0}^{n} a_i N^i$$

for some constants a_i, $i \in \{0, \dots, n\}$. The SQP performs a number of QP iterations, and the iteration count is limited by $\bar{i}_{\text{SQP}}$. Therefore, the upper-bound on the SQP solution time can be modelled by

$$\zeta_{\text{SQP}}(\mathbf{p}) \coloneqq \bar{i}_{\text{SQP}} \sum_{i=0}^{n} a_i N^i \tag{6.2}$$

In the simulations of this study, the QP iteration limit is set at $\bar{i}_{\text{SQP}} = 15$. A representative result for the relationship between the solution time and number of prediction steps N for a range of sampling time h is shown in Fig. 6.1. The QP solution time ζ is slightly different to that obtained in (4.20) from Fig. 4.7. This is due to the different size of the model used in for the missile engagement (5.12) and the and PAA (4.21) test cases, the former with 11 and the latter with 5 states. Nonetheless, the average solution time ζ for each QP solved in each of the 10 trials is generally increasing with N with mostly linear relationship. This is consistent with Assumption 3.6. From the graph, the model for QP solution time is chosen as $\zeta = (0.8N + 4) \times 10^{-3}$ such that

$$\zeta_{\text{SQP}}(\mathbf{p}) = 15\,(0.8N + 4) \times 10^{-3}$$

as per Proposition 6.1

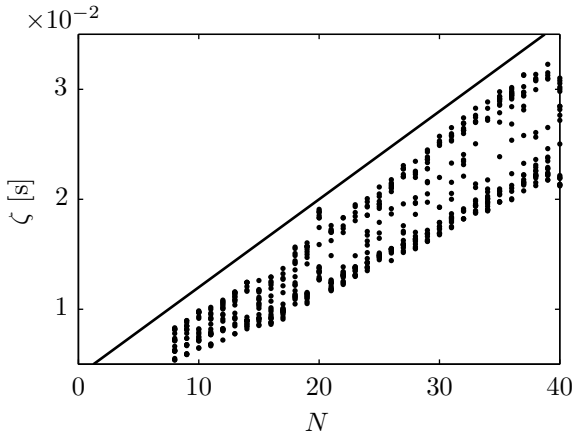

Figure 6.1: Relationship between the average solution time per QP iteration against number of prediction steps N for 10 trials for the OCP (5.16). The solid line depicts the model chosen for the solution time ζ.

6.3 Multi-objective design of iMAG

In this section, the multi-objective design of MPC (MOD-MPC) formulated in Chapter 3 will be revisited. For the study, the closed-loop system of interest involves a missile controlled by the model-predictive integrated control (iMAG) proposed in Chapter 5 to intercept a target in a specified engagement scenario. The approach aims to aid the practitioner in designing the controller by providing insights on the optimal trade-off between control performance and required computational capacity. Consistent with Chapters 3 and 4, this study focuses on the calibration of structural design parameters of the control algorithm, namely the sampling time h and prediction horizon length N. These parameters, denoted by $\mathbf{p} := (h, N)$, affect both design objectives. The parameter choice leads to the competitive pairing between the two design objectives, following Lemma 4.10, so that improvement of one objective by calibration of the design parameters comes at the compromise of the other.

Given the closed-loop system of a missile in an engagement with a target, control performance is measured by miss-distance (6.1) that is based on the control law (5.17) with the OCP (5.16). Monotonicity properties of the miss-distance with respect to the design parameters can be identified similar to that of the closed-loop value function in Propositions 4.2 and 4.3.

Assumption 6.1 (Non-monotonicity of Δ with respect to h and N)**.** *Consider the control performance $\Delta(\mathsf{x}_0, \cdot)$ as in* (6.1)*. For $\mathbf{p} \coloneqq (h, N)$, $h \mapsto V^*(\mathcal{X}_0, \mathbf{p})$ is generally non-monotonic with respect to h for a given N, and generally non-monotonic with respect to N for a given h*

The required computational capacity can be measured based on the time taken by a processor to solve the OCP (5.16). $\zeta_{\text{SQP}}(\mathbf{p})$ in (6.2) is the upper-bound of this time for the simulation hardware for the design parameter $\mathbf{p}$. Subsequently, the non-dimensional *utilisation number* (3.27) is used to indicate the computational power of the implementation hardware, relative to that used in simulation, that is required for the implementation of the MPC design. This metric is given by

$$\eta_{\text{SQP}}(\mathbf{p}) \coloneqq \zeta_{\text{SQP}}(\mathbf{p})/h. \tag{6.3}$$

The utilisation number is a design objective that will be minimised. Based on the choice of $\mathbf{p}$, minimizing the required computational capacity implies reduction of the cost of the control hardware to be implemented onboard the missile.

Theorem 6.1. *The objective functions $\Delta(\mathsf{x}_0, \cdot)$ and $\eta_{SQP}(\cdot)$ are competing as per Definition 1.6 for the design parameter $\mathbf{p} \coloneqq (h, N)$ given that Proposition 6.1 and Assumption 6.1 hold.*

Proof. The proof closely follows that of Lemma 4.10. Given the non-monotonicity of Δ from Assumption 6.1 and monotonicity of η_{SQP} as per Proposition 6.1 and (6.3), there must exist some set $\mathcal{P}_\text{c}$ so that the two objective functions are competing. A shorter sampling time in and/or longer prediction horizon can potentially improve closed-loop control performance (decrease Δ) as the control receives more frequent feedback and capture more future dynamics of the plant, respectively. However, the faster sampling rate and/or longer prediction horizon required more computational load (increase η_{SQP}). The reverse is also true, a delayed sampling time and shorter prediction horizon worsens performance and reduces computational burden. □

The multi-objective co-design of MPC (MOD-MPC) problem can be formulated following (3.28),

$$\mathcal{P}_\bullet(\mathcal{P}_\text{s}) \coloneqq \arg\,\text{m-}\min_{\mathbf{p}}\ \boldsymbol{\ell}(\mathbf{p}) \tag{6.4a}$$

$$\text{s.t.}\ \ \mathbf{p} \in \mathcal{P}_\text{s}. \tag{6.4b}$$

The two objectives are contained in the objective vector $\boldsymbol{\ell}(\mathbf{p}) = (\Delta(\mathbf{p}), \eta_{\text{SQP}}(\mathbf{p}))$. $\mathcal{P}_\text{s}$ defines the search space and $\mathcal{P}_\bullet$ defines the set of the design parameters that have an optimal trade-off, that is the Pareto optimal solution. The set of objective evaluations $\mathcal{L} := \{\boldsymbol{\ell}(\mathbf{p}) | \mathbf{p} \in \mathcal{P}_\bullet\}$ shows the trade-off curve of the two objectives. These set contain Pareto optimal points that are non-dominated as per Definitions 1.4 and 1.5. That is, a point $\boldsymbol{\ell}(\mathbf{p}_\bullet)$ where $\mathbf{p}_\bullet \in \mathcal{P}$ is a Pareto optimal in $\mathcal{P}$ if there does *not* exist another design choice $\mathbf{p} \in \mathcal{P}$ such that $\boldsymbol{\ell}(\mathbf{p}) \prec \boldsymbol{\ell}(\mathbf{p}_\bullet)$.

6.4 Results

The formulated MOD-MPC problem (6.4) is solved to demonstrate the design process of the proposed missile control system. Missile specifications, initial conditions, constraints and disturbance, as well as OCP formulation and solver are as specified in Section 5.3.1. The optimiser algorithm used is the Dividing Triangles (DITRI) algorithm proposed in Section 4.4.2, motivated by its effectiveness and efficiency in solving the MOD-MPC problem by performing a focused search and is guaranteed to converge to the solution for a given search space. The bounded search space is $\mathcal{P}_\text{s} = \{(h, N) | h \in [0.005, 0.03], N \in [6, 15]\}$.

Two sets of initial conditions are considered, denoted respectively as engagement scenario 1 and 2 in Section 5.3.1. Fig. 6.2 shows the solution of the MOD-MPC problem with the disturbance (5.19) for the first engagement scenario. A trade-off curve is obtained after 20 evaluation points in DITRI. There are 6 different designs in the resulting optimal design set from which a practitioner can choose from. For example, the practitioner can choose design 20 with a sampling time of around 26 ms and 8 prediction steps. Based on the result, the average miss-distance would be around 1.2 m. From the value of utilisation number η_{SQP}, this particular design choice would require around 6 times the speed achieved by the simulation hardware. Similar conclusions can be made from the results for the second engagement scenario given in Fig. 6.3. For instance, design 14 requires around $\eta_{\text{SQP}} = 8$ to achieve 1.5 m miss-distance with $h = 0.025$ and $N = 12$. The simulation results of control design 14 is given in Fig. 6.4.

In Fig. 6.2, it is observed that the resulting trade-off curve flattens at just over 1 m average miss-distance after around utilisation number $\eta_{\text{SQP}} = 5$ for the first engagement scenario. This point is slightly different for the second scenario,

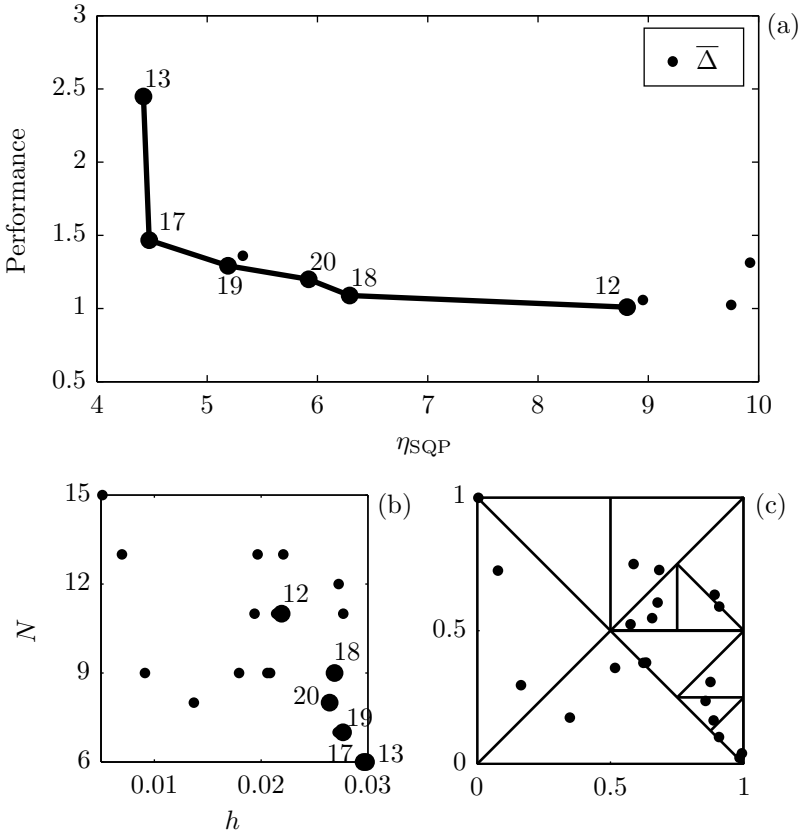

Figure 6.2: Pareto optimal solution for the design of the iMAG controller in engagement scenario 1. Evaluation points are plotted as dots with larger dots indicating Pareto optimality[1].

where performance starts to flattens at around $\eta_{\mathrm{SQP}} = 6$. At this point, performance comes close to the maximum, after which an increase in computational capacity does not significantly improve control performance. Such an insight provides a guidance in the necessary cost to implement the controller. The different results between the two engagement scenarios highlight that insights drawn from the results are dependent on the choice of initial conditions that represents the intended closed-loop control conditions. More conservative designs are obtained from the more challenging scenario 2. For example, a miss distance be-

[1]Subplot (a): the resulting solution in the objective space, making up the trade-off curve between performance, $\overline{\Delta}$ (black) or $\overline{\Delta} + \sigma_{\Delta}$ (grey), and required capacity η_{SQP}. Subplots (b) and (d): the associated solution in parameter $\mathbf{p}$ space for both the mean (b) and deviated (d) performance metrics. Subplots (c) and (e): the accompanying plot showing evaluation points in normalised space and triangle divisions used internally in DITRI. Total number of evaluations is $\overline{i^{\mathrm{ev}}} = 20$.

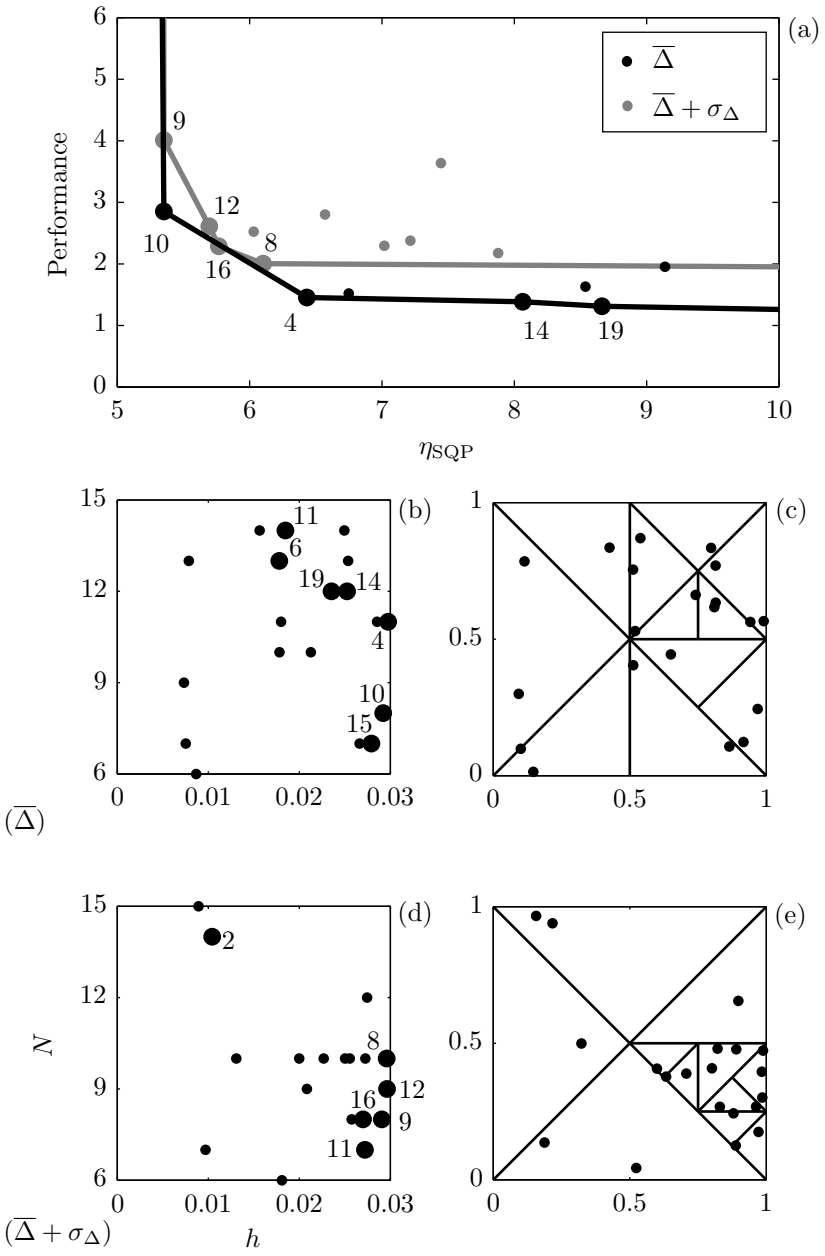

Figure 6.3: Pareto optimal solution for the design of the iMAG controller in engagement scenario 2. Evaluation points are plotted as dots with larger dots indicating Pareto optimality[1].

low 1.5 m for the second engagement scenario is achieved with at least around $\eta_{\mathrm{SQP}} = 6.5$ instead of $\eta_{\mathrm{SQP}} = 4.5$ for the first scenario.

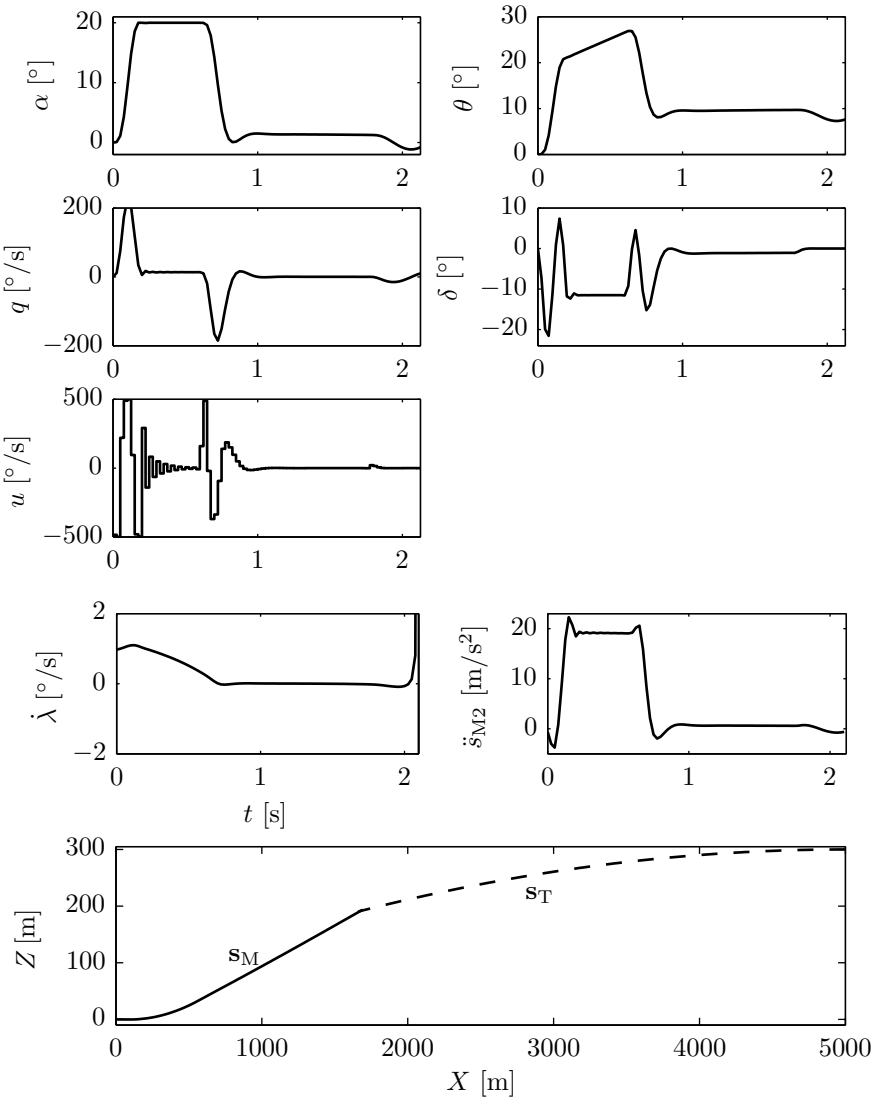

Figure 6.4: Evolution of dynamic and kinematic states for engagement scenario 2. $h = 0.025$, $N = 12$.

A plot (shaded) showing the miss-distance 1σ away from the mean is given to accompany the main result in Fig. 6.3. This can be used to explore Pareto optimal designs when smaller distributions of miss-distance (performance robustness under disturbance) are prioritised more relative to nominal miss-distance. As expected, the associated trade-off curve closely resembles that of the mean, but shifted up in miss-distance as shown in Fig. 6.2(a). To competitively minimise this metric (instead of the mean) smaller values of prediction steps N are

now Pareto optimal, whilst optimal sampling time values h remain similar as before, as shown in Fig. 6.2(d). This concludes that a reduced prediction horizon length is more competitive in optimizing the required computational load than control performance, when control performance is measured by $\overline{\Delta} + \sigma_{\Delta}$.

Finally, note that the overall required computational capacity for the obtained design choices is considerably large, ranging from a utilisation number of around $\eta_{\text{SQP}} = 5$ to 10. This highlights the high computational cost associated with MPC, particularly as a non-linear MPC is considered in this study. To functionally implement the controller, there are a number of techniques that can be used to address the high computational cost by increasing the capacity of the implementation hardware, including the use of compiled programming language, parallel-processing, an increased clock-frequency, and pipelining.

6.5 Conclusions

In this chapter, multi-objective tuning of the proposed model-predictive integrated missile control system from Chapter 5 was demonstrated based on the MOD-MPC approach from Chapter 3. The first objective function specifies control performance and is measured by the miss distance of the missile when it intercepts the target. The second objective is the required computational capacity to implement the controller formulated based on the non-linear MPC used for the proposed integrated controller. The two objective functions are extensions of the formulation given in Chapter 4, where the closed-loop value function and computational capacity for a linear MPC were used to indicate performance and implementation cost, respectively. Simulation results show that the competing relationship between performance and required capacity remains, with an associated Pareto front that defines the optimal trade-off between the two objectives, upon which a practitioner can draw insights from to assist with design decisions. In overall, this chapter demonstrated the MOD-MPC approach on a closed-loop system with non-linear MPC, i.e. the model-predictive integrated missile control, and a non-linear controlled plant representing an engagement – completing the investigations in Chapter 4 for linear MPC with linear and non-linear plants – to conclude the main body of this thesis.

Chapter 7
Contributions and further work

7.1 Contributions to MPC design approach

A multi-objective design approach for model-predictive control (MPC) has been outlined in the thesis. The development and subsequent analysis of the design scheme in Chapters 3 and 4 address the first and second research aims that were summarised at the end of Chapter 2. This section recounts the main contributions of the two chapters to the literature.

Multi-objective design of closed-loop performance and required computational capacity

The literature review has highlighted that required computational capacity is an underlying design objective in MPC calibration alongside control performance. This is primarily due to the fact that MPC is typically associated with high levels of computational load. The required hardware capability to implement MPC in real-time prescribes the overall implementation cost. Existing investigations focus mainly on the control performance both in single- [GS10] or multi-objective fashion [RMBSM14], but often neglects required computational capacity or treats it as a design constraint. Such an isolated approach could potentially lead to implementation infeasibility and sub-optimality issues that extend the design cycle as the algorithm and implementation designs are treated separately. Moreover, in the studies involving model-based controllers, the open-loop value function is often used as the measure for performance, e.g. [BB14], which is not always ideal since control is applied in a closed-loop feedback system.

Chapter 3 of the thesis formulated a design approach for MPC with a multi-objective outlook optimising both control performance based on closed-loop simulations and required computational capacity to implement the MPC. Performance is measured by metrics that are based on closed-loop simulations of the controlled plant, whilst the utilisation number is used to measure required computational capacity in multiples of the capability of the simulation hardware. Pertinent design parameters were outlined as those that are associated with the control algorithm and affect both objectives, denoted as the structural parame-

ters. The proposed design approach was cast as a multi-objective optimisation problem, whose associated solution is the Pareto front that defines the optimal trade-off relationship between control performance and required computational capacity. From this solution, the practitioner can draw conclusions such as the optimal control performance for a range of computational capacity, as well as the required hardware capability for a unit change in performance based on the gradient of the trade-off curve. This allows for a more streamlined design approach that avoids sub-optimal or infeasible designs.

Analytical results for the multi-objective design of MPC

Although multi-objective optimisation has been highlighted as an effective design approach for autonomous systems [AH14, Ker14], discussions in MPC are yet to be fully explored. Chapter 4 analysed the multi-objective design approach for MPC that was proposed in Chapter 3. Analytical results in the chapter provide theoretical contributions concerning the nature of the open- and closed-loop objective functions of the MPC system. In particular, smoothness properties of the pertaining value functions were presented with respect to structural design parameters, namely the sampling time and prediction horizon length, which are often treated as constant [BKS14]. This extends previous studies on the performance tuning via such design parameters, for example [BB14]. Subsequently, the competitive pairing of the two design objectives of control performance and required computational capacity were formalised to form the basis of the development of a targeted design optimiser.

An effective and efficient multi-objective optimiser

Based on the lemmas and theorems presented in Chapter 4, several necessary and sufficient conditions for an effective and efficient solver of the proposed multi-objective design of MPC problem were prescribed. Based on the prescribed conditions, an optimiser that can solve the proposed MOD-MPC approach accurately and quickly can be developed. This is important as objective evaluations in the design approach involve closed-loop simulations that can be time-consuming, especially if multiple scenarios are considered, for instance, the consideration of different initial conditions and control trajectories. The chapter concluded by developing an effective and efficient solver that satisfies the prescribed condi-

tions, whose performance was subsequently compared against existing standard solvers from [DPAM02, TSC13]. Consequently, the sufficiency of the prescribed conditions for effectiveness and efficiency was validated.

7.2 Contributions to missile control

As discussed in the literature review, the integration of autopilot and guidance in missile control has been shown to lead to an improved performance [MO01a]. Several studies have followed the proposed integrated system with the developments based on linear control strategies [MSO03, PJ99, SIG06]. The dynamics associated with the autopilot and guidance subsystems are non-linear and often constrained. Therefore, the integration of the two subsystems warrants for an advanced controller that can directly address the resulting difficult dynamics.

Consequently, a model-predictive integrated missile control system was developed in Chapter 5. The proposed control scheme was then tested in a simulative study of an engagement between a controlled missile and a manoeuvring target. Comparison was made with a separated missile control with the augmented proportional navigation guidance and an ideal autopilot. The results show that the proposed control is more accurate and responsive compared to the conventional scheme, potentially suitable for high-performing missiles. In closing, the missile control was designed using the proposed multi-objective design of MPC approach in Chapter 6. As part of the demonstration, a measure for control performance was proposed based on the miss-distance of the missile at the point of interception. Furthermore, the model-predictive control proposed uses a non-linear prediction model. The investigation of the closed-loop system with non-linear MPC and non-linear plant in Chapter 6 completes the studies of Chapter 4 where a linear MPC was used for both linear and non-linear plant, to address the third and final research aim.

7.3 Further studies

7.3.1 Consideration of other design parameters

The thesis focussed on the structural design parameters of sampling time and prediction horizon length as the parameters that are calibrated in optimising control performance and required computational capacity. A potential extension is

to consider other coupled design parameters, which are those that affect both design objectives, including those that are associated with the implementation design of the controller. This presents a logical, but non-trivial, potential work to supplement the thesis.

Prediction model type as a design parameter

Structural design parameters are those that are coupled and associated with the control algorithm design. Prediction model type is the only pertinent structural design parameter that was excluded as a design parameter to be calibrated in this thesis. Although the thesis considered different types of prediction model, i.e. linear time-invariant (LTI) and non-linear models, the investigations are done individually. In Chapter 4, the study was done with an LTI prediction model, whilst the study with an OCP subject to a non-linear prediction model was done in Chapter 6 for the proposed model-predictive integrated missile control system.

A logical extension of this thesis would be to consider prediction model type as a design parameter. The extension would need to further analyse the MOD-MPC approach for a closed-loop system with a non-linear MPC. Although numerical results presented in Chapter 6 were consistent with that obtained for linear MPC, complete analytical results were not presented to comprehensively support the findings. This warrants an analysis to extend the theory presented in Chapter 4 to include those associated with a non-linear prediction model.

Hardware design parameters

Although the sampling time and prediction horizon length dictate the implementation of the MPC, they are not associated exclusively to implementation design. Parameters that are directly associated with the implementation aspect of MPC include the solver algorithm, solver tolerance and numerical precision used. Inclusion of these parameters presents a full co-design scheme for the software and hardware of MPC as an extension to the MOD-MPC approach in this thesis. Such an approach will follow previous useful studies in the topic, including [SLKC13].

7.3.2 Calibration of more complex MPC formulations

Many MPC developments rely on the addition of constraints and other cost function forms in the OCP. For instance, terminal state constraint can be used to guar-

antee stability [MRRS00, MR09], whilst a more general cost function may be formulated to consider an *economic* performance instead of tracking error [DAR11, FRLC10]. The multi-objective design approach developed in this thesis can be adopted for the design of such more extensive MPC formulations.

7.3.3 Investigation of other case studies

The case studies that were investigated in the thesis include a pitch autopilot and engine control in Chapter 4, as well as a missile-target engagement control in Chapter 6. For each of the cases, a set of initial conditions and control trajectories was specified for the closed-loop simulation of the system. Although a number of different scenarios were considered, it certainly did not represent the whole operation range of the controlled plant. Consequently, a potential extension to the case studies considered in the thesis is to consider more initial conditions and/or control trajectories.

Further than the consideration of a more extensive range of initial conditions and control trajectories, there are countless of other plant models that could be studied. Although the chosen missile and engine control examples are real-world scenarios adopted or developed from current literature, a more extensive set of case studies is a potential further work that can demonstrate the value of the proposed MOD-MPC scheme. For instance, the missile-target engagement control studied simulates the situation in a planar space. A full three-dimensional study could better reflect the scenario and evaluate control performance more accurately. A case study extension could also consider a scenario that is entirely different from those studied in the thesis, in defence applications or otherwise.

Appendix A

Mathematical formulations

A.1 The OCP in sparse representation

A sparse representation of the dQ-Q-OCP in (3.16) keeps the state and input as optimisation variables instead of using a composite variable as, for instance, $\mathbf{z}$ in (4.6). First, define the optimisation variable

$$\mathbf{z}_\mathrm{s} \coloneqq (u_0, x_1, u_1, x_2, \ldots, u_{N-1}, x_N, u_N, x_\mathrm{t}).$$

The concatenated cost function weight is given by

$$H_\mathrm{s} \coloneqq \operatorname{diag}\left(\overline{R}, \begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}, \begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}, \ldots, \begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}, Q_\mathrm{t}\right)$$

The prediction model can be rewritten as follows

$$\begin{bmatrix} \overline{B} & -I_{n_\mathrm{x}} & & & & & & \\ & \overline{A} & \overline{B}_1 & -I_{n_\mathrm{x}} & & & & \\ & & & \ddots & & & & \\ & & & \overline{A} & \overline{B} & -I_{n_\mathrm{x}} & & \\ & & & & & \overline{A}_\mathrm{r} & \overline{B}_\mathrm{r} & -I_{n_\mathrm{x}} \end{bmatrix} \mathbf{z}_\mathrm{s} = - \begin{bmatrix} \overline{A}\mathsf{x}_i \\ \mathbf{0}_{n_\mathrm{x}\times 1} \\ \vdots \\ \mathbf{0}_{n_\mathrm{x}\times 1} \\ \mathbf{0}_{n_\mathrm{x}\times 1} \end{bmatrix}$$

$$G_\mathrm{s}\mathbf{z}_\mathrm{s} = W_\mathrm{s}.$$

Consequently, this yields the sparse representation in standard form,

$$\mathbf{z}_\mathrm{s}^* \coloneqq \arg\min_{\mathbf{z}_\mathrm{s}} \|\mathbf{z}_\mathrm{s}\|_{H_\mathrm{s}}^2 \tag{A.1a}$$

$$\text{s.t.} \quad G_\mathrm{s}\mathbf{z}_\mathrm{s} = W_\mathrm{s} \tag{A.1b}$$

$$\underline{\mathbf{z}}_\mathrm{s} \leq \mathbf{z}_\mathrm{s} \leq \overline{\mathbf{z}}_\mathrm{s}. \tag{A.1c}$$

If the chosen numerical solver is unable to handle lower- and upper-bounds (A.1c) directly, slack variables can be used to make the constraints (A.1b) and (A.1c) uniform as either equality or inequality constraints.

A.2 Augmented PN with ideal autopilot

Proportional navigation (PN) is obtained from the unconstrained regulation of the terminal distance between . Consider the kinematics of the engagement as illustrated in Fig. 5.2. A linear-quadratic regulation can be formulated as the following

$$(s_2^*, v_2^*, a_{\mathrm{T2}}^*) \coloneqq \arg \min_{(s_2, v_2, a_{\mathrm{T2}})} \frac{1}{2} \left\| \begin{bmatrix} s_2(t_{\mathrm{go}}) \\ v_2(t_{\mathrm{go}}) \\ a_{\mathrm{T2}}(t_{\mathrm{go}}) \end{bmatrix} \right\|_{Q_{\mathrm{APN}}}^2 + \frac{1}{2} \int_0^{t_{\mathrm{go}}} a_{\mathrm{M2}}^2 \, d\tau$$

$$\text{s.t. } \frac{d}{dt} \begin{bmatrix} s_2 \\ v_2 \\ a_{\mathrm{T2}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_2 \\ v_2 \\ a_{\mathrm{T2}} \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} a_{\mathrm{M2}}$$

to minimise the terminal engagement state at time t_{go} based on the cost weight Q_{APN}. The intercept time t_{go} (or an estimate) is assumed to be available. For an intercept, the terminal distance between the missile and target is to be minimised to zero, which can be achieved by setting the cost weight as follows

$$Q_{\mathrm{APN}} = \mathrm{diag}\,(q_{\mathrm{APN}}, 0, 0) \qquad \forall q_{\mathrm{APN}} > 0.$$

The optimal control can be defined at the limit where a zero terminal distance s_2 is infinitely weighted, yielding the APN guidance law

$$a_{\mathrm{APN}} \coloneqq \lim_{q_{\mathrm{APN}} \to \infty} a_{\mathrm{M2}}^* = \frac{3}{t_{\mathrm{go}}^2} \left(s_2 + v_2 t_{\mathrm{go}} + \frac{1}{2} a_{\mathrm{T2}} t_{\mathrm{go}}^2 \right)$$

as given in [PBL10b].

Assuming an ideal autopilot, the commanded acceleration $a_{\mathrm{APN}}(t)$ is the achieved (vertical) acceleration of the missile at any time t during the engagement as the dynamics of the missile is ignored.

Appendix B
Numerical methods

B.1 Sequential quadratic programming

The discrete-time optimal control problem (OCP) in (3.9) can be solved using a sequential quadratic programming (SQP) algorithm. To do so, the prediction model is recast as a linear time-varying (LTV) model for the algorithm to iteratively solve towards the original non-linear model. Note that dependence on the design parameters $\mathbf{p}$ are not stated throughout the section to keep a succinct notation.

First, the Jacobians and a new offset vector at each prediction step k are defined as follow

$$\hat{A}_k = \left.\frac{\partial \overline{f}}{\partial x}\right|_{\hat{x}_k, \hat{u}_k} \qquad\qquad \hat{B}_k = \left.\frac{\partial \overline{f}}{\partial u}\right|_{\hat{x}_k, \hat{u}_k}$$

and

$$\hat{F}_k(\hat{x}_k, \hat{u}_k) = \overline{f}(\hat{x}_k, \hat{u}_k) - \hat{A}_k \hat{x}_k - \hat{B}_k \hat{u}_k.$$

The newly defined states $\hat{\mathbf{x}} := (\hat{x}_0, \ldots, \hat{x}_N)$ and inputs $\hat{\mathbf{u}} := (\hat{u}_0, \ldots, \hat{u}_{N-1})$ are estimates of the solution $(\mathbf{x}^*, \mathbf{u}^*)$ of the OCP. The differences between the predictive and estimated states and inputs can now be captured in the following equality constraint

$$x_{k+1} = \hat{A}_k x_k + \hat{B}_k u_k + \hat{F}_k \qquad\qquad \forall k \in \{0, \ldots, N-1\} \tag{B.1}$$

or in matrix form

$$\begin{bmatrix} \hat{B}_0 & -I_{n_x} & & & & \\ \hat{A}_1 & \hat{B}_1 & -I_{n_x} & & & \\ & & \ddots & & & \\ & & & \hat{A}_{N-1} & \hat{B}_{N-1} & -I_{n_x} \end{bmatrix} \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ \vdots \\ u_{N-1} \\ x_N \end{bmatrix} = - \begin{bmatrix} \hat{F}_0 + \hat{A}_0 x_0 \\ \hat{F}_1 \\ \vdots \\ \hat{F}_{N-1} \end{bmatrix}.$$

Algorithm 3 SQP

Require: Current state x_i, (optional) hotstarted estimates $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$

1: Set tolerance $\epsilon_x \ll 1$, $\epsilon_u \ll 1$ and $\bar{i}_{\text{QP}}$
2: **if** not hotstarted **then**
3: Initialise the estimates
$\hat{x}_k \leftarrow \mathsf{x}_i \mathbf{1}_{n_\mathsf{x} \times 1}, \forall k \in \{0, \dots, N\}$
$\hat{u} \leftarrow \mathbf{0}_{n_\mathsf{u} \times 1}, \forall k \in \{0, \dots, N-1\}$
4: **end if**
5: $i_{\text{QP}} \leftarrow 0$
6: **loop**
7: Solve the QP (B.2) given $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$
to obtain the solution $(\hat{\mathbf{x}}^*, \hat{\mathbf{u}}^*)$
8: $i_{\text{QP}} \leftarrow i_{\text{QP}} + 1$
9: Compute normalised errors (Δ_x, Δ_u)
10: **if** $\Delta_x < \epsilon_x$ **and** $\Delta_u < \epsilon_u$, **or** $i_{\text{QP}} > \bar{i}_{\text{QP}}$ **then**
11: Terminate loop
12: **else**
13: Re-initialise
$\hat{x}_k \leftarrow \hat{x}_k^*, \forall k \in \{0, \dots, N\}$
$\hat{u}_k \leftarrow \hat{u}_k^*, \forall k \in \{0, \dots, N-1\}$
14: **end if**
15: **end loop**

The equality defines an LTV model that replaces (3.9c) to reduce the OCP to a QP with a linear time-varying prediction model,

$$(\mathbf{x}^*, \mathbf{u}^*) := \arg\min_{(\mathbf{x},\mathbf{u})} \sum_{k=0}^{N-1} \left\| \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\|^2_{\begin{bmatrix} \overline{Q} & \overline{S} \\ \overline{S}^\mathsf{T} & \overline{R} \end{bmatrix}} + \|x_N\|^2_{\overline{Q}_\mathsf{t}} \tag{B.2a}$$

$$\text{s.t.} \quad x_0 = \mathsf{x}_i \tag{B.2b}$$

$$x_{k+1} = \hat{A}_k x_k + \hat{B}_k u_k + \hat{F}_k \qquad \forall k \in \{0, \dots, N-1\} \tag{B.2c}$$

$$x_k \in [\underline{x}, \overline{x}]\,,\ u_k \in [\underline{u}, \overline{u}] \qquad \forall k \in \{0, \dots, N-1\}. \tag{B.2d}$$

Following [TDD10], the SQP algorithm to solve the dOCP is given in Algorithm 3. In Step 7, the solver used for the QP is assumed to be convergent. Examples are given in [BV04]. When well-initialised, the algorithm imposes that

$$(\hat{\mathbf{x}}^*, \hat{\mathbf{u}}^*) \simeq (\mathbf{x}^*, \mathbf{u}^*).$$

Consequently, from (B.1)

$$x^*_{k+1} = \overline{f}(x^*_k, u^*_k) \qquad \forall k \in \{0, \dots, N-1\}$$

to satisfy (3.9c).

Proposition B.1 (Convergence of Algorithm 3 [TDD10]). *$(\hat{\mathbf{x}}^*, \hat{\mathbf{u}}^*)$ in Algorithm 3 converges to $(\mathbf{x}^*, \mathbf{u}^*)$ if $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ is initialised sufficiently close to $(\mathbf{x}^*, \mathbf{u}^*)$.*

In Step 9, an example error measure is a normalised mean squares of the state and inputs,

$$\Delta_x = \frac{1}{N+1} \sum_{k=0}^{N} ((\hat{x}^*_k - \hat{x}_k)/\overline{x})^2 \qquad \Delta_u = \frac{1}{N} \sum_{k=0}^{N-1} ((\hat{u}^*_k - \hat{u}_k)/\overline{u})^2$$

where $\overline{x}$ and $\overline{u}$ are the upper-bounds on the states and inputs respectively. In Step 10, the condition $i_{\mathrm{QP}} > \bar{i}_{\mathrm{QP}}$ is used to limit the solution time of the routine.

Appendix C
Proofs

C.1 Proof of Lemma 4.3

This proof will show that the Lagrange multipliers can be expressed in the form

$$\boldsymbol{\mu}^+ = \boldsymbol{\mu}^* + \partial_h \mu \delta_h + \partial_T \mu \delta_T + \mathcal{O}(\delta_h^2, \delta_T^2) \tag{C.1}$$

for some finite $\partial_h \mu$ and $\partial_T \mu$, showing the differentiability of $\boldsymbol{\mu}$. Consider a perturbation $h^+ = h + \delta_h$ as $\delta_h \to 0$.

Case 1 ($\mathcal{A}(\mathbf{z}^*) = \mathcal{A}(\mathbf{z}^+)$)

If the active set stays the same, the perturbed KKT conditions have a solution of $\acute{\boldsymbol{\mu}}^+ = -\big((\acute{G} + \partial_h \acute{G}\delta_h)(H + \partial_h H \delta_h)^{-1}(\acute{G} + \partial_h \acute{G}\delta_h)^\mathsf{T}\big)^{-1}\big(\acute{W} + (\acute{Y} + \partial_h \acute{Y}\delta_h)\mathsf{x}_i\big) = \acute{\boldsymbol{\mu}}^* + \partial_h \acute{\mu}\delta_h + \mathcal{O}(\delta_h^2)$ for some finite $\partial_h \acute{\mu}$, instead of (4.8). This is organised as follows,

$$\begin{aligned}\boldsymbol{\mu}^+ &= \mathcal{I}_{\mathcal{A}(\mathbf{z}^+)}\acute{\boldsymbol{\mu}}^+ = \mathcal{I}_{\mathcal{A}(\mathbf{z}^*)}\left(\acute{\boldsymbol{\mu}}^* + \partial_h \acute{\mu}\delta_h + \mathcal{O}(\delta_h^2)\right) \\ &= \boldsymbol{\mu}^* + \partial_h \mu \delta_h + \mathcal{O}(\delta_h^2).\end{aligned}$$

Case 2 – Entering constraint ($\mathcal{A}(\mathbf{z}^*) \subset \mathcal{A}(\mathbf{z}^+)$)

Consider a perturbation causing a constraint $e \notin \mathcal{A}(\mathbf{z}^*)$ to be active at some point along the perturbation, $\mathcal{A}(\mathbf{z}^*) \subset \mathcal{A}(\mathbf{z}^+) = \mathcal{A}(\mathbf{z}^*) \cup \{e\}$. The additional multipliers $\acute{\boldsymbol{\mu}}^+$ as constraint e activates are

$$\begin{aligned}\acute{\boldsymbol{\mu}}^+ = \begin{bmatrix}\acute{\mu}_o^+ \\ \acute{\mu}_e^+\end{bmatrix} &= \left(\left\|\begin{bmatrix}\acute{G} + \partial_h \acute{G}\delta_h \\ \acute{G}_e^+\end{bmatrix}^\mathsf{T}\right\|^2_{(H+\partial_h H\delta_h)^{-1}}\right)^{-1}\begin{bmatrix}\acute{W} + (\acute{Y} + \partial_h \acute{Y}\delta_h)\mathsf{x}_i \\ \acute{W}_e^+ + \acute{Y}_e^+ \mathsf{x}_i\end{bmatrix} + \mathcal{O}(\delta_h^2) \\ &= \begin{bmatrix}\acute{G}H^{-1}\acute{G}^\mathsf{T} & \acute{G}H^{-1}{\acute{G}_e^+}^\mathsf{T} \\ \acute{G}_e^+ H^{-1}\acute{G}^\mathsf{T} & \acute{G}_e^+ H^{-1}{\acute{G}_e^+}^\mathsf{T}\end{bmatrix}^{-1}\begin{bmatrix}\acute{W} + \acute{Y}\mathsf{x}_i \\ \acute{W}_e^+ + \acute{Y}_e^+\mathsf{x}_i\end{bmatrix} + \begin{bmatrix}\partial_h \acute{\mu}_o \\ \partial_h \acute{\mu}_e\end{bmatrix}\delta_h + \mathcal{O}(\delta_h^2)\end{aligned} \tag{C.2}$$

for some finite $\partial_h \acute{G}$, $\partial_h \acute{\mu}_o$ and $\partial_h \acute{\mu}_e$. The subscript e denotes rows of $\acute{G}, \acute{W}$, etc. associated with the entering constraint, and o for the original constraint set. Let

$$\begin{bmatrix} \acute{G}H^{-1}\acute{G}^\mathsf{T} & \acute{G}H^{-1}\acute{G}_e^{+\mathsf{T}} \\ \acute{G}_e^{+}H^{-1}\acute{G}^\mathsf{T} & \acute{G}_e^{+}H^{-1}\acute{G}_e^{+\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \mathcal{G}_1 & \mathcal{G}_2 \\ \mathcal{G}_3 & \mathcal{G}_4 \end{bmatrix} =: \mathcal{G}$$

and the Schur's complement $\mathcal{G}_\mathrm{S} := \mathcal{G}_4 - \mathcal{G}_3\mathcal{G}_1^{-1}\mathcal{G}_2$. The solution to (C.2) can be found first by block inversion of $\mathcal{G}$ [1].

First, the lower block can be expressed as

$$\begin{aligned} \acute{\mu}_e^+ &= -\mathcal{G}_\mathrm{S}^{-1}\mathcal{G}_3\mathcal{G}_1^{-1}\left(\acute{W} + \acute{Y}\mathsf{x}_i\right) + \mathcal{G}_\mathrm{S}^{-1}\left(\acute{W}_e^+ + \acute{Y}_e^+\mathsf{x}_i\right) + \partial_h\acute{\mu}_e\delta_h + \mathcal{O}(\delta_h^2) \\ &= -\mathcal{G}_\mathrm{S}^{-1}\left(\mathcal{G}_3\mathcal{G}_1^{-1}(\acute{W} + \acute{Y}\mathsf{x}_i) - \acute{W}_e^+ - \acute{Y}_e^+\mathsf{x}_i\right) + \partial_h\acute{\mu}_e\delta_h + \mathcal{O}(\delta_h^2). \end{aligned}$$

After some algebra, expansion of the bracketed terms above would reveal that

$$\begin{aligned} \mathcal{G}_3\mathcal{G}_1^{-1}(\acute{W} + \acute{Y}\mathsf{x}_i) - \acute{W}_e^+ - \acute{Y}_e^+\mathsf{x}_i &= \acute{G}_e^+H^{-1}\acute{G}^\mathsf{T}(\acute{G}H^{-1}\acute{G}^\mathsf{T})^{-1}\left(\acute{W} + \acute{Y}\mathsf{x}_i\right) - \acute{W}_e^+ - \acute{Y}_e^+\mathsf{x}_i \\ &= -\acute{G}_e^+H^{-1}\acute{G}^\mathsf{T}\acute{\boldsymbol{\mu}}^* - \acute{W}_e^+ - \acute{Y}_e^+\mathsf{x}_i \\ &= \left(\acute{G}_e + \partial_h G_e\delta_h\right)\mathbf{z}^* - \acute{W}_e - \left(\acute{Y}_e + \partial_h Y_e\delta_h\right)\mathsf{x}_i + \mathcal{O}(\delta_h^2) \\ &= c_e(\mathbf{z}^*) + \partial_h\mathcal{G}_e\delta_h + \mathcal{O}(\delta_h^2) \\ &= 0 + \partial_h\mathcal{G}_e\delta_h + \mathcal{O}(\delta_h^2). \end{aligned}$$

This consistent with the fact that $\mu_e^* = 0$ since the constraint e is inactive in the unperturbed OCP i.e. $c_e(\mathbf{z}^*) < 0$. Substitution yields the Lagrange multiplier for the newly active constraint $\acute{\mu}_e^+ = \left(\partial_h\acute{\mu}_e + \partial_h\mathcal{G}_e\right)\delta_h + \mathcal{O}(\delta_h^2)$.

For the upper block, values are obtained by substituting (4.8) and the bracket expansion above after the blockwise inversion, yielding

$$\begin{aligned} \acute{\mu}_o^+ &= \left(\mathcal{G}_1^{-1} + \mathcal{G}_1^{-1}\mathcal{G}_2\mathcal{G}_\mathrm{S}^{-1}\mathcal{G}_3\mathcal{G}_1^{-1}\right)(\acute{W} + \acute{Y}\mathsf{x}_i) \\ &\qquad - \mathcal{G}_1^{-1}\mathcal{G}_2\mathcal{G}_\mathrm{S}\left(\acute{W}_e^+ + \acute{Y}_e^+\mathsf{x}_i\right) + \partial_h\mu_o\delta_h + \mathcal{O}(\delta_h^2) \\ &= \mathcal{G}_1^{-1}(\acute{W} + \acute{Y}\mathsf{x}_i) + \left(\partial_h\mu_o + \partial_h\mathcal{G}_o\right)\delta_h + \mathcal{O}(\delta_h^2) \\ &= \left(\acute{G}H^{-1}\acute{G}^\mathsf{T}\right)^{-1}(\acute{W} + \acute{Y}\mathsf{x}_i) + \left(\partial_h\mu_o + \partial_h\mathcal{G}_o\right)\delta_h + \mathcal{O}(\delta_h^2) \\ &= \acute{\boldsymbol{\mu}}^* + \left(\partial_h\acute{\mu}_o + \partial_h\mathcal{G}_o\right)\delta_h + \mathcal{O}(\delta_h^2). \end{aligned}$$

[1] $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1}$
$= \begin{bmatrix} (A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$.

Substituting these into (C.2),

$$\acute{\boldsymbol{\mu}}^{+} = \begin{bmatrix} \acute{\boldsymbol{\mu}}^{*} \\ 0 \end{bmatrix} + \begin{bmatrix} \partial_h^{\acute{\mu}_o} + \partial_h \mathcal{G}_o \\ \partial_h \acute{\mu}_e + \partial_h^{\mathcal{G}_e} \end{bmatrix} \delta_h + \mathcal{O}(\delta_h^2)$$

yields

$$\boldsymbol{\mu}^{+} = \mathcal{I}_{\mathcal{A}(\mathbf{z}^{+})} \acute{\boldsymbol{\mu}}^{+} = \boldsymbol{\mu}^{*} + \Big(\mathcal{I}_{\mathcal{A}(\mathbf{z}^{*})} (\partial_h \acute{\mu}_o + \partial_h \mathcal{G}_o) + \mathcal{I}_{\{e\}} (\partial_h \acute{\mu}_e + \partial_h \mathcal{G}_e) \Big) \delta_h + \mathcal{O}(\delta_h^2).$$

Case 3 – Exiting constraint ($\mathcal{A}(\mathbf{z}^{*}) \supset \mathcal{A}(\mathbf{z}^{+})$)

To keep the working tidy, suppose that the exiting constraint is located at the end of the set $\mathcal{A}(\mathbf{z}^{*})$. The perturbed solution is now $\acute{\boldsymbol{\mu}}^{+} = \mathcal{I}_{\mathcal{A}(\mathbf{z}^{*})} \big(\| (\acute{G} + \partial_h G \delta_h)^{\mathsf{T}} \|^2_{(H + \partial_h H \delta_h)^{-1}} \big)^{-1} \big(\acute{W} + (\acute{Y} + \partial_h^{Y} \delta_h) \mathsf{x}_i \big) - \mathcal{I}_{\{e\}} \big(\| (\acute{G}_e + \partial_h G_e \delta_h)^{\mathsf{T}} \|^2_{(H + \partial_h H \delta_h)^{-1}} \big)^{-1} \big(\acute{W}_e + (\acute{Y}_e + \partial_h Y_e \delta_h) \mathsf{x}_i \big)$, where the subtractive term is to make the Lagrange multipliers associated with the exiting constraint zero. Further expansion yields

$$\begin{aligned} \acute{\boldsymbol{\mu}}^{+} &= \mathcal{I}_{\mathcal{A}(\mathbf{z}^{*})} \big(\acute{G} H^{-1} \acute{G}^{\mathsf{T}} \big)^{-1} \big(\acute{W} + \acute{Y} \mathsf{x}_i \big) - \\ &\quad \mathcal{I}_{\{e\}} \big(\acute{G}_e H^{-1} \acute{G}_e^{\mathsf{T}} \big)^{-1} \big(\acute{W}_e + \acute{Y}_e \mathsf{x}_i \big) + \partial_h \mu \delta_h + \mathcal{O}(\delta_h^2) \\ &= \begin{bmatrix} \mathcal{I}_{\mathcal{A}(\mathbf{z}^{+})} & 0 \end{bmatrix} \acute{\boldsymbol{\mu}}^{*} + \partial_h \acute{\mu} \delta_h + \mathcal{O}(\delta_h^2). \end{aligned}$$

C.2 Proof of Lemma 4.4

Case 1 – Constant N

Consider a perturbation $h^{+} := h + \delta_h$ as $\delta_h \to 0$ and that, over the perturbation, the number of prediction steps remains constant. Subject to this, $H(h^{+}) = H(h) + \partial_h^{H} \delta_h + \mathcal{O}(\delta_h^2)$ and similarly for L, G, W and Y as these matrices consist of addition, multiplication and/or inversion of the differentiable $\overline{A}$, $\overline{B}, \overline{Q}, \overline{R}, \overline{S}, \overline{Q}_{\mathrm{r}}, \overline{R}_{\mathrm{r}}$ and/or $\overline{S}_{\mathrm{r}}$. The perturbed OCP is

$$\mathbf{z}^{*+}(\mathsf{x}_i) = \arg\min_{\mathbf{z}} \ \ell_z^{+}(\mathbf{z}, h^{+}) \tag{C.3a}$$

where $\ell_z^{+}(\mathbf{z}, h^{+}) := \|\mathbf{z}\|^2_{H + \partial_h H \delta_h} + \|\mathsf{x}_i\|^2_{L + \partial_h L \delta_h} + \mathcal{O}(\delta_h^2) = \ell_z(\mathbf{z}) + \partial_h \ell_z(\mathbf{z}) \delta_h + \mathcal{O}(\delta_h^2)$. Subject to

$$0 \geq \mathbf{c}^{+}(\mathbf{z}^{*+}) = (G + \partial_h G \delta_h) \mathbf{z}^{*+} - W - (Y + \partial_h Y \delta_h) \mathsf{x}_i + \mathcal{O}(\delta_h^2) \tag{C.3b}$$

with a value function $J_z^+ = \ell_z^+(\mathbf{z}^{*+})$. The perturbed solution is $\mathbf{z}^{*+} = -(H + \partial_h H \delta_h)^{-1}(G + \partial_h G \delta_h)^\mathsf{T} \boldsymbol{\mu}^+ + \mathcal{O}(\delta_h^2)$. Substituting (C.1) yields a solution of the form $\mathbf{z}^{*+} = \mathbf{z}^* + \partial_h z \delta_h + \mathcal{O}(\delta_h^2)$. A similar sequence of derivations can be followed for T, $T^+ = T + \delta_T$, affecting $\overline{Q}_\mathrm{r}$, $\overline{R}_\mathrm{r}$ and $\overline{S}_\mathrm{r}$ and yielding the perturbed system of the form (C.3). The solution can consequently be expressed as

$$\mathbf{z}^{*+} = \mathbf{z}^* + \partial_h \mathbf{z}^* \delta_h + \partial_T z \delta_T + \mathcal{O}(\delta_h^2, \delta_T^2) \tag{C.4}$$

for some finite $\partial_h z$ and $\partial_T z$. Hence, $\mathbf{z}^*$ is differentiable.

Case 2 – Changing N

Now consider a perturbation $h^+ := h + \delta_h$ as $0 < \delta \to 0$ applied to an OCP with $N - 1$ prediction steps and $h_\mathrm{r} = h$. The unperturbed OCP has the following matrices related to (4.3)–(4.5),

$$\widetilde{\mathbf{B}} = \begin{bmatrix} \widetilde{\underline{B}} \\ \widetilde{B}_\bullet \end{bmatrix} \qquad \widetilde{\mathbf{Q}} = \begin{bmatrix} \widetilde{\underline{Q}} & \\ & Q_f \end{bmatrix} \qquad \mathbf{R} = \underline{R} \qquad \mathcal{R} = \underline{\mathcal{R}}.$$

Now consider a perturbation $h^- = h - \delta_h$ as $0 < \delta \to 0$ on an OCP with N prediction steps and $h_\mathrm{r} = 0$. The unperturbed OCP now has

$$\widetilde{\mathbf{B}} = \begin{bmatrix} \widetilde{\underline{B}} & 0 \\ \widetilde{B}_\bullet & 0 \\ \widetilde{B}_\bullet & 0 \end{bmatrix} \quad \widetilde{\mathbf{Q}} = \begin{bmatrix} \widetilde{\underline{Q}} & & \\ & 0 & \\ & & Q_f \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \underline{R} & \\ & 0 \end{bmatrix} \quad \mathcal{R} = \begin{bmatrix} \underline{\mathcal{R}} & \\ & 0 \end{bmatrix}.$$

After some algebra, it can be shown that the solution $\mathbf{z}^* = -H^{-1}G^\mathsf{T}\boldsymbol{\mu}$ associated with these block matrices are

$$\begin{aligned} \mathbf{z}^*_{(1)} &= \alpha \begin{bmatrix} (I - \mathcal{R}\widetilde{\underline{B}}) & -(I - \mathcal{R}\widetilde{\underline{B}}) & \widetilde{\underline{B}} & \widetilde{B}_\bullet & -\widetilde{\underline{B}} & -\widetilde{B}_\bullet \end{bmatrix} \boldsymbol{\mu} \\ \mathbf{z}^*_{(2)} &= \begin{bmatrix} \alpha \begin{bmatrix} (I - \mathcal{R}\widetilde{\underline{B}}) & 0 & -(I - \mathcal{R}\widetilde{\underline{B}}) & 0 & \widetilde{\underline{B}} & \widetilde{B}_\bullet & \widetilde{B}_\bullet & -\widetilde{\underline{B}} & -\widetilde{B}_\bullet & -\widetilde{B}_\bullet \end{bmatrix} \\ 0 \end{bmatrix} \boldsymbol{\mu} \end{aligned} \tag{C.5}$$

respectively, where $\alpha := -2\left(\widetilde{\underline{B}}^\mathsf{T}\widetilde{\underline{Q}}\widetilde{\underline{B}} + \widetilde{B}_\bullet^\mathsf{T} Q_f \widetilde{B}_\bullet + \underline{R}\right)$. The two equations in (C.5) yields the same solution (the residual step in $\mathbf{z}^*_{(2)}$ contributes only addition of zeroes). However, the rate of change over the perturbation is not.

For the OCP with N prediction steps and $h_\mathrm{r} = 0$, the rate of change of $\mathbf{z}^*$ w.r.t. h can be obtained from $\partial_h \mathbf{z}^*_{(2)}$ before the substitution $h_\mathrm{r} = 0$. This would have additional terms compared to $\partial_h \mathbf{z}^*_{(1)}$ of that with $N - 1$ steps. These terms are

associated with the constraints for the additional prediction step and will be zero only if the additional constraints are inactive so that the Lagrange multipliers and their rates of change are zero. Hence, the solution cannot be expressed as (C.4) in general, since $\partial_h \mathbf{z}^*$ is not guaranteed to be unique and differentiability of $\mathbf{z}^*$ at changing N is not guaranteed. A similar result is obtained when looking at the rate of change w.r.t. T.

Bibliography

[AAT08] R. Amari, M. Alamir, and P. Tona. Unified MPC strategy for idle speed control, vehicle start-up and gearing applied to an automated manual transmission. In *Proc. IFAC World Congress*, pp. 7079–7085, 2008.

[Adl56] F. P. Adler. Missile guidance by three-dimensional proportional navigation. *J. of Applied Physics*, 27(5):500–507, 1956.

[AdSC12] H. V. H. Ayala and L. dos Santos Coelho. Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications*, 39(10):8968–8974, 2012.

[AH14] J. T. Allison and D. R. Herber. Multidisciplinary design optimization of dynamic engineering systems. *AIAA J.*, 52(4):691–710, 2014.

[APPI10] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International J. of Vehicle Autonomous Systems*, 8(2-4):190–216, 2010.

[ATA09] R. Amari, P. Tona, and M. Alamir. Experimental evaluation of a hybrid MPC strategy for vehicle start-up with an automated manual transmission. In *Proc. IEEE European Control Conf.*, pp. 4271–4277, 2009.

[BB14] E. Bini and G. M. Buttazzo. The optimal sampling pattern for linear control systems. *IEEE Trans. on Automatic Control*, 59(1):78–90, 2014.

[BBBM08] M. Baotic, F. Borrelli, A. Bemporad, and M. Morari. Efficient online computation of constrained optimal control. *SIAM J. Control and Optimization*, 47(5):2470–2489, 2008.

[BC08] E. Bini and A. Cervin. Delay-aware period assignment in control systems. In *IEEE Real-Time Systems Symposium, 2008*, pp. 291–300, 2008.

[BDNB08] E. Bini, M. Di Natale, and G. Buttazzo. Sensitivity analysis for fixed-priority real-time systems. *Real-Time Systems*, 39(1-3):5–30, 2008.

[BFFB12] G. A. Bunin, F. Fraire, G. François, and D. Bonvin. Run-to-run MPC tuning via gradient descent. *Computer Aided Chemical Engineering*, 30:927–931, 2012.

[BKMM16] V. Bachtiar, E. C. Kerrigan, W. H. Moase, and C. Manzie. Continuity and monotonicity of the MPC value function with respect to sampling time and prediction horizon. *Automatica*, 63:330–337, 2016.

[BKS14] P. Bagheri and A. Khaki-Sedigh. An analytical tuning approach to multivariable model predictive controllers. *J. of Process Control*, 24(12):41–54, 2014.

[BM93] P. Bendotti and M. M'Saad. A skid-to-turn missile autopilot design: The generalized predictive adaptive control approach. *International J. of Adaptive Control and Signal Processing*, 7(1):13–31, 1993.

[BMBH15] T. Broomhead, C. Manzie, M. Brear, and P. Hield. Model reduction of diesel mean value engine models. *SAE Technical Paper*, 2015.

[BMDP02] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[BMM+14] V. Bachtiar, T. Mühlpfordt, W. Moase, T. Faulwasser, R. Findeisen, and C. Manzie. Nonlinear model predictive missile control with a stabilising terminal constraint. In *Proc. IFAC World Congress*, pp. 457–462, 2014.

[BMMK16a] V. Bachtiar, C. Manzie, W. Moase, and E. C. Kerrigan. Analytical results for the multi-objective design of model predictive control. *Control Engineering Practice*, 2016. In press.

[BMMK16b] V. Bachtiar, C. Manzie, W. Moase, and E. C. Kerrigan. Multi-objective co-design of a model-based integrated missile control. *AIAA J. of Guidance, Control, and Dynamics*, 2016. Under review.

[BSMD08] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans. on Evolutionary Computation*, 12(3):269–283, 2008.

[BTW12] S. N. Balakrishnan, A. Tsourdos, and B. A. White. *Advances in missile guidance, control, and estimation*, Vol. 5. CRC Press, 2012.

[BV04] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[BWG91] R. P. Bitmead, V. Wertz, and M. Gerers. *Adaptive optimal control: the thinking man's GPC*. Prentice Hall, 1991.

[CA98] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[CB04] E. F. Camacho and C. Bordons. *Model predictive control*, Vol. 2 of *Advanced textbooks in control and signal processing*. Springer, 2004.

[CBG03] W.-H. Chen, D. J. Ballance, and P. J. Gawthrop. Optimal control of nonlinear systems: a predictive control approach. *Automatica*, 39(4):633–641, 2003.

[CCL02] C. A. Coello Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proc. IEEE Congress on Evolutionary Computation*, pp. 1051–1056, 2002.

[CDK03] M. Cannon, V. Deshmukh, and B. Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39(8):1487–1494, 2003.

[CGS84] S. W. Chan, G. C. Goodwin, and K. S. Sin. Convergence properties of the Riccati difference equation in optimal filtering of nonstabilizable systems. *IEEE Trans. on Automatic Control*, 29(2):110–118, 1984.

[CMVV11] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. on Optimization*, 21(3):1109–1140, 2011.

[Cot71] R. G. Cottrell. Optimal intercept guidance for short-range tactical missiles. *AIAA J.*, 9(7):1414–1415, 1971.

[DAR11] M. Diehl, R. Amrit, and J. B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Trans. on Automatic Control*, 56(3):703–707, 2011.

[Deb01] K. Deb. *Multi-objective optimization using evolutionary algorithms*, Vol. 16. John Wiley & Sons, 2001.

[DHS01] E. Devaud, J.-P. Harcaut, and H. Siguerdidjane. Three-axes missile autopilot design: from linear to nonlinear control strategies. *AIAA J. of Guidance, Control, and Dynamics*, 24(1):64–71, 2001.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6(2):182–197, 2002.

[EKSH09] R. Eaton, J. Katupitiya, K. W. Siew, and B. Howarth. Autonomous farming: Modelling and control of agricultural machinery in a unified framework. *International J. of intelligent systems technologies and applications*, 8(1-4):444–457, 2009.

[ET10] V. Exadaktylos and C. J. Taylor. Multi-objective performance optimisation for model predictive control by goal attainment. *International J. of Control*, 83(7):1374–1386, 2010.

[Fle01] E. L. Fleeman. *Tactical missile design*. American Institute of Aeronautics and Astronautics, 2001.

[FRLC10] A. Ferramosca, J. B. Rawlings, D. Limón, and E. F. Camacho. Economic MPC for a changing economic criterion. In *Proc. IEEE Conf. on Decision and Control*, pp. 6131–6136, 2010.

[FSF97] V. Fromion, G. Scorletti, and G. Ferreres. Nonlinear performance of a pi controlled missile: an explanation. In *Proc. IEEE Conf. on Decision and Control*, pp. 4198–4203, 1997.

[GD15] M. Guay and D. Dochain. A multi-objective extremum-seeking controller design technique. *International J. of Control*, 88(1):38–53, 2015.

[GKM10] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J. of Intelligent and Robotic Systems*, 57(1-4):65–100, 2010.

[GPM89] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practicea survey. *Automatica*, 25(3):335–348, 1989.

[GQD12] S. Gros, R. Quirynen, and M. Diehl. Aircraft control based on fast non-linear MPC & multiple-shooting. In *Proc. IEEE Conf. on Decision and Control*, pp. 1142–1147, 2012.

[GS10] J. L. Garriga and M. Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.

[Gue76] M. Guelman. The closed-form solution of true proportional navigation. *IEEE Trans. on Aerospace and Electronic Systems*, (4):472–482, 1976.

[Gur13] Gurobi Optimization. Gurobi optimizer reference manual, 2013. http://www.gurobi.com.

[Ha96] Q. Ha. Robust sliding mode controller with fuzzy tuning. *IET Electronics Letters*, 32(17):1626–1628, 1996.

[HC07] X.-B. Hu and W.-H. Chen. Model predictive control for non-linear missiles. *Proc. IMechE Pt. I: J. of Systems and Control Engineering*, 221(8):1077–1089, 2007.

[HPT+12] J. Holdren, T. Power, G. Tassey, A. Ratcliff, and L. Christodoulou. A national strategic plan for advanced manufacturing. 2012.

[Int15] Intel Corporation. Intel® core™i7-3770 Processor (8M Cache, up to 3.90 GHz), 2015. http://ark.intel.com/products/65719/.

[ISG07] M. Idan, T. Shima, and O. M. Golan. Integrated sliding mode autopilot-guidance for dual-control missiles. *AIAA J. of Guidance, Control, and Dynamics*, 30(4):1081–1089, 2007.

[Jac10] P. B. Jackson. Overview of missile flight control systems. *J. Hopkins APL Tech D*, 29(1), 2010.

[JMK14] G. A. N. Júnior, M. A. Martins, and R. Kalid. A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty. *ISA Trans.*, 53(2):560–567, 2014.

[Jon01] D. R. Jones. DiRect global optimization algorithm. In *Encyclopedia of Optimization*, pp. 431–440. 2001.

[JPS93] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *J. of Optimization Theory and Applications*, 79(1):157–181, 1993.

[KC02] M. Kale and A. Chipperfield. Reconfigurable flight control strategies using model predictive control. In *Proc. IEEE International Symposium on Intelligent Control*, pp. 43–48, 2002.

[Ker14] E. C. Kerrigan. Co-design of hardware and algorithms for real-time optimization. In *Proc. IEEE European Control Conf.*, pp. 2484–2489, 2014.

[Kha02] H. K. Khalil. *Nonlinear systems*. Prentice Hall, 3rd edition, 2002.

[KKK00] K. B. Kim, M.-J. Kim, and W. H. Kwon. Receding horizon guidance laws with no information on the time-to-go. *AIAA J. of Guidance, Control, and Dynamics*, 23(2):193–199, 2000.

[KKS04] S.-H. Kim, Y.-S. Kim, and C. Song. A robust adaptive nonlinear control approach to missile autopilot design. *Control engineering practice*, 12(2):149–154, 2004.

[KM00] E. C. Kerrigan and J. M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proc. IEEE Conf. on Decision and Control*, pp. 4951–4956, 2000.

[KTK80] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.

[KWK16] J.-H. Kim, I. H. Whang, and B. M. Kim. Finite horizon integrated guidance and control for terminal homing in vertical plane. *AIAA J. of Guidance, Control, and Dynamics*, 39(5):1104–1112, 2016.

[KYK01] K. B. Kim, T.-W. Yoon, and W. H. Kwon. Receding horizon guidance laws for constrained missiles with autopilot lags. *Control Engineering Practice*, 9(10):1107–1115, 2001.

[LKC14] S. Longo, E. C. Kerrigan, and G. A. Constantinides. Constrained LQR for low-precision data representation. *Automatica*, 50(1):162–168, 2014.

[LSA71] A. H. Levis, R. A. Schlueter, and M. Athans. On the behaviour of optimal linear sampled-data regulators. *International J. of Control*, 13(2):343–361, 1971.

[Mac02] J. M. Maciejowski. *Predictive control with constraints*. Prentice Hall, 2002.

[May14] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[MHL99] M. Morari and J. H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[MK71] S. M. Melzer and B. C. Kuo. Sampling period sensitivity of the optimal sampled data linear regulator. *Automatica*, 7(3):367–370, 1971.

[MM90] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Trans. on Automatic Control*, 35(7):814–824, 1990.

[MM91] H. Michalska and D. Q. Mayne. Receding horizon control of nonlinear systems without differentiability of the optimal value function. *Systems & control letters*, 16(2):123–130, 1991.

[MM93] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. on Automatic Control*, 38(11):1623–1633, 1993.

[MO01a] P. Menon and E. Ohlmeyer. Integrated design of agile missile guidance and autopilot systems. *Control Engineering Practice*, 9(10):1095–1106, 2001.

[MO01b] P. Menon and E. J. Ohlmeyer. Nonlinear integrated guidance-control laws for homing missiles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 6–8, 2001.

[MR09] D. Q. Mayne and B. Rawlings. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, 2000.

[MSO03] P. Menon, G. Sweriduk, and E. Ohlmeyer. Optimal fixed-interval integrated guidance-control laws for hit-to-kill missiles. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, 2003.

[MSOM04] P. Menon, G. Sweriduk, E. Ohlmeyer, and D. Malyevac. Integrated guidance and control of moving-mass actuated kinetic warheads. *AIAA J. of Guidance, Control, and Dynamics*, 27(1):118–126, 2004.

[MTB14] M. Mahmoodabadi, M. Taherkhorsandi, and A. Bagheri. Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO. *Neurocomputing*, 124:194–209, 2014.

[NN84] F. W. Nesline and M. L. Nesline. How autopilot requirements constrain the aerodynamic design of homing missiles. In *Proc. IEEE American Control Conf.*, pp. 716–730, 1984.

[NRR93] R. A. Nichols, R. T. Reichert, and W. J. Rugh. Gain scheduling for H_∞ controllers: a flight control example. *IEEE Trans. on Control Systems Technology*, 1(2):69–79, 1993.

[NZ81] F. W. Nesline and P. Zarchan. A new look at classical vs modern homing missile guidance. *AIAA J. of Guidance, Control, and Dynamics*, 4(1):78–85, 1981.

[Ohl96] E. J. Ohlmeyer. Root-mean-square miss distance of proportional navigation missile against sinusoidal target. *AIAA J. of Guidance, Control, and Dynamics*, 19(3):563–568, 1996.

[PBL10a] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd. Basic principles of homing guidance. *Johns Hopkins APL Technical Digest*, 29(1):25–41, 2010.

[PBL10b] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd. Modern homing missile guidance theory and techniques. *Johns Hopkins APL Technical Digest*, 29(1):42–59, 2010.

[PJ99] N. F. Palumbo and T. D. Jackson. Integrated missile guidance and control: A state dependent riccati differential equation approach. In *Proc. IEEE International Conf. on Control Applications*, pp. 243–248, 1999.

[PZ95] T. Parisini and R. Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443–1451, 1995.

[QB03] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[Rei92] R. T. Reichert. Dynamic scheduling of modern-robust-control autopilot designs for missiles. *IEEE Control Systems*, 12(5):35–42, 1992.

[RM93] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Trans. on Automatic Control*, 38(10):1512–1516, 1993.

[RMBSM14] G. Reynoso-Meza, X. Blasco, J. Sanchis, and M. Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 28:58–73, 2014.

[RMGNSB13] G. Reynoso-Meza, S. García-Nieto, J. Sanchis, and F. X. Blasco. Controller tuning by means of multi-objective optimization algorithms: a global tuning framework. *IEEE Trans. on Control Systems Technology*, 21(2):445–458, 2013.

[RP01] S.-H. Ryu and J.-H. Park. Auto-tuning of sliding mode control parameters using fuzzy logic. In *Proc. IEEE American Control Conf.*, Vol. 1, pp. 618–623, 2001.

[RU97] K. Y. Rani and H. Unbehauen. Study of predictive controller tuning methods. *Automatica*, 33(12):2243–2248, 1997.

[Rui04] J. P. Ruiz. *Reconfigurable autopilot design for a high performance aircraft using model predictive control*. PhD thesis, Massachusetts Institute of Technology, 2004.

[SC93] J. S. Shamma and J. R. Cloutier. Gain-scheduled missile autopilot design using linear parameter varying transformations. *AIAA J. of Guidance, Control, and Dynamics*, 16(2):256–263, 1993.

[SC98] R. Shridhar and D. J. Cooper. A tuning strategy for unconstrained multivariable model predictive control. *Industrial & Engineering Chemistry Research*, 37(10):4003–4016, 1998.

[Sch94] D. A. Schumacher. *Tactical missile autopilot design using nonlinear control*. PhD thesis, University of Michigan, 1994.

[SDBP14] U. K. Sahu, P. N. Dwivedi, A. Bhattacharya, and R. Padhi. Modified generalized explicit guidance for midcourse with near-zero lateral acceleration in terminal phase. In *Proc. IFAC World Congress*, pp. 3954–3959, 2014.

[SE11] G. Shah and S. Engell. Tuning MPC for desired closed-loop performance for MIMO systems. In *Proc. IEEE American Control Conf.*, pp. 4404–4409, 2011.

[She12] R. Shekhar. *Variable Horizon Model Predictive Control: Robustness & Optimality*. PhD thesis, University of Cambridge, 2012.

[SIG06] T. Shima, M. Idan, and O. M. Golan. Sliding-mode control for integrated missile autopilot guidance. *AIAA J. of Guidance, Control, and Dynamics*, 29(2):250–260, 2006.

[Sio04] G. M. Siouris. *Missile guidance and control systems*. Springer, 2004.

[SLKC13] A. Suardi, S. Longo, E. C. Kerrigan, and G. A. Constantinides. Energy-aware mpc co-design for dc-dc converters. In *Proc. IEEE European Control Conf.*, pp. 3608–3613, 2013.

[SM90] U. S. Shukla and P. R. Mahapatra. The proportional navigation dilemma-pure or true? *IEEE Trans. on Aerospace and Electronic Systems*, 26(2):382–392, 1990.

[SMR99] P. O. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. on Automatic Control*, 44(3):648–654, 1999.

[ST09] Y. B. Shtessel and C. H. Tournes. Integrated higher-order sliding mode guidance and autopilot for dual control missiles. *AIAA J. of Guidance, Control, and Dynamics*, 32(1):79–94, 2009.

[Ste11] J. Stewart. *Calculus*. Cengage Learning, 7th edition, 2011.

[TC12] W.-Q. Tang and Y.-L. Cai. Predictive functional control-based missile autopilot design. *AIAA J. of Guidance, Control, and Dynamics*, 35(5):1450–1455, 2012.

[TDD10] Q. Tran-Dinh and M. Diehl. Local convergence of sequential convex programming for nonconvex optimization. In *Recent Advances in Optimization and its Applications in Engineering*, pp. 93–102. 2010.

[TGKP11] S. Talole, A. Godbole, J. Kolhe, and S. Phadke. Robust roll autopilot design for tactical missiles. *AIAA J. of Guidance, Control, and Dynamics*, 34(1):107–117, 2011.

[TI98] A. Thukral and M. Innocenti. A sliding mode missile pitch autopilot synthesis for high angle of attack maneuvering. *IEEE Trans. on Control Systems Technology*, 6(3):359–371, 1998.

[TMTCV14] M. Taherkhorsandi, M. Mahmoodabadi, M. Talebipour, and K. Castillo-Villar. Pareto design of an adaptive robust hybrid of PID and sliding control for a biped robot via genetic algorithm optimization. *Nonlinear Dynamics*, 79(1):251–263, 2014.

[TPB00] W. Tan, A. K. Packard, and G. J. Balas. Quasi-LPV modeling and LPV control of a generic missile. In *Proc. IEEE American Control Conf.*, Vol. 5, pp. 3692–3696, 2000.

[TSC13] M. Tesch, J. Schneider, and H. Choset. Expensive multiobjective optimization for robotics. In *Proc. IEEE International Conf. on Robotics and Automation*, pp. 973–980, 2013.

[vdLSY08] J. van der Lee, W. Svrcek, and B. Young. A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making. *ISA Trans.*, 47(1):53–59, 2008.

[VFT08] P. Vega, M. Francisco, and F. Tadeo. Multiobjective optimization for automatic tuning of robust model based predictive controllers. In *Proc. IFAC World Congress*, 2008.

[VSB+88] S. R. Vukelich, S. L. Stoy, K. A. Burns, J. A. Castillo, and M. E. Moore. Missile DATCOM Final report, vol. 1. Technical Report AFWAL-TR-86-3091, McDonnell Douglas Missile Systems Company, 1988.

[WIHM08] L. Wang, H. Ishida, T. Hiroyasu, and M. Miki. Examination of multi-objective optimization method for global search using DIRECT and GA. In *Proc. IEEE Congress on Evolutionary Computation*, pp. 2446–2451, 2008.

[WN99] S. Wright and J. Nocedal. *Numerical optimization*. Springer, 1999.

[XLG10] Y. Xue, D. Li, and F. Gao. Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. *Control Engineering Practice*, 18(1):67–76, 2010.

[Yan08] R. Yanushevsky. *Modern missile guidance*. CRC Press, 2008.

[YC92] P.-J. Yuan and J.-S. Chern. Ideal proportional navigation. *AIAA J. of Guidance, Control, and Dynamics*, 15(5):1161–1165, 1992.

[Zar99] P. Zarchan. Ballistic missile defense guidance and control issues. *Science & Global Security*, 8(1):99–124, 1999.

[Zar12] P. Zarchan. *Tactical and strategic missile guidance, progress in astronautics and aeronautics*. AIAA, 2012.

[ZJM11] M. N. Zeilinger, C. N. Jones, and M. Morari. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE Trans. on Automatic Control*, 56(7):1524–1534, 2011.

[ZN42] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64(11), 1942.