Optimisation and Control of Aerospace Vehicles

Martin Doff-Sotta

Queen's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2023

Abstract

Convex optimisation is concerned with reformulating nonlinear optimisation problems into convex programs which are computationally tractable and whose solution is globally optimal. When leveraged in single-shot optimisation or multi-stage optimisation problems such as model predictive control, convex programs can accelerate computation tremendously, paving the way to online implementations of algorithms that were long thought to be intractable. This is particularly attractive in the context of safety-critical aerospace applications where a reliable solution is often required in real time.

This thesis investigates the use of convex optimisation and model predictive control in emerging applications from green aviation and urban air mobility. This work is motivated by the need to decarbonise the air transport industry in order to meet the requirements of net zero transport.

At first we consider the problem of energy management for a hybrid-electric aircraft. Through a convex formulation of mathematical models of the propulsion system, a computationally tractable optimisation problem is constructed whose globally optimal solution is used to arbitrate in real time the power demand of the aircraft between the gas turbine and electric motor, allowing significant fuel savings. Computation times are also reduced by an order of magnitude thanks to a custom implementation of a first order solver, enabling fast real-time implementations for new generations of electric and hybrid aircraft.

We then explore the use of difference of convex functions (DC) decomposition to compute optimal trajectories for the transition of a tiltwing vertical take-off and landing (VTOL) aircraft in the presence of uncertainty. A DC decomposition of the dynamics is computed and convexity is exploited to bound uncertain system trajectories tightly, allowing to define a robust optimisation for air transport of people in urban areas.

Finally, building on the theory of DC functions decomposition, a systematic data-driven approach to obtain computationally tractable robust model predictive control (MPC) algorithms for nonlinear systems has been developed and applied to VTOL aircraft in urban air mobility scenarios. The resulting control scheme offers robustness guarantees to model uncertainty and exogenous disturbances, laying the foundation for future widespread adoption in safety-critical applications related to decarbonisation of the transport sector, green aviation and urban air mobility.

Optimisation and Control of Aerospace Vehicles

Martin Doff-Sotta

Queen's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

Acknowledgements

First of all, I would like to thank my supervisors Prof Mark Cannon and Dr Marko Bacic for their invaluable advice throughout my DPhil project and their time helping me advance my career.

Marko, I will never forget your phone call a few days before Christmas in 2018 to let me know you would recommend me for admission at Oxford, this has changed my life forever. Thank you for trusting me to conduct this research for Rolls-Royce and for the funding you brought to the project. Thank you also for your expertise in everything aerospace-related, this has improved the quality of the project tremendously in terms of its applicability in the real world.

Mark, thank you for being such an awesome supervisor and for the healthy work environment you have set within the group. Working with you, I felt trusted and empowered. You gave me the freedom to shape this project and conduct my research independently. Your availability throughout the project was remarkable, along with your impressive expertise in model predictive control and optimisation. You helped establish a sound theoretical foundation for my thesis.

I would like to also acknowledge the generous funding I received from Rolls-Royce, UKRI, and the Fernand Lazard foundation to conduct this research.

Throughout my journey in academia, I have met many inspirational people, some helped me become who I am today. If I could name but a few, I would mention Prof James Richard Forbes from McGill University, Canada who gave me my first research job and with whom I co-authored my first research paper in 2018, an experience that undoubtedly brought me closer to Oxford.

Oxford being such an intellectually stimulating place, I would like to thank the people that made my journey so memorable and with whom I shared countless interesting conversations. These include Manos, Sam, Soufiane, Jean-Louis, my colleagues from the department, friends from Queen's College and many others.

Finally, I would like to thank my family and close friends without whom this adventure would not have been possible. I am thankful to my parents for their support during this journey and for making me feel at home when I am back to Belgium. A huge thanks also goes to Sohmal for her love, kindness and all the delicious meals she cooked when I was busy writing this thesis.

Abstract

Convex optimisation is concerned with reformulating nonlinear optimisation problems into convex programs which are computationally tractable and whose solution is globally optimal. When leveraged in single-shot optimisation or multi-stage optimisation problems such as model predictive control, convex programs can accelerate computation tremendously, paving the way to online implementations of algorithms that were long thought to be intractable. This is particularly attractive in the context of safety-critical aerospace applications where a reliable solution is often required in real time.

This thesis investigates the use of convex optimisation and model predictive control in emerging applications from green aviation and urban air mobility. This work is motivated by the need to decarbonise the air transport industry in order to meet the requirements of net zero transport.

At first we consider the problem of energy management for a hybrid-electric aircraft. Through a convex formulation of mathematical models of the propulsion system, a computationally tractable optimisation problem is constructed whose globally optimal solution is used to arbitrate in real time the power demand of the aircraft between the gas turbine and electric motor, allowing significant fuel savings. Computation times are also reduced by an order of magnitude thanks to a custom implementation of a first order solver, enabling fast real-time implementations for new generations of electric and hybrid aircraft.

We then explore the use of difference of convex functions (DC) decomposition to compute optimal trajectories for the transition of a tiltwing vertical take-off and landing (VTOL) aircraft in the presence of uncertainty. A DC decomposition of the dynamics is computed and convexity is exploited to bound uncertain system trajectories tightly, allowing to define a robust optimisation for air transport of people in urban areas.

Finally, building on the theory of DC functions decomposition, a systematic data-driven approach to obtain computationally tractable robust model predictive control (MPC) algorithms for nonlinear systems has been developed and applied to VTOL aircraft in urban air mobility scenarios. The resulting control scheme offers robustness guarantees to model uncertainty and exogenous disturbances, laying the foundation for future widespread adoption in safety-critical applications related to decarbonisation of the transport sector, green aviation and urban air mobility.

Contents

List of Figures

List of Abbreviations

ADMM	Alternating direction method of multipliers
CDCS	Charge depleting charge sustaining
DC	Difference of convex functions
DCNN	Difference of convex neural network
DC-TMPC	Difference of convex tube-based model predictive control
DP	Dynamic programming
EOM	Equations of motion
ICNN	Input-convex neural network
LMI	Linear matrix inequality
LQR	Linear quadratic regulator
LTV	Linear time variant
LTI	Linear time invariant
MAE	Mean absolute error
MPC	Model predictive control
MSE	Mean squared error
MRE	Mean relative error
MTOW	Maximum take-off weight
NLP	Nonlinear program
NMPC	Nonlinear model predictive control
NTMPC	Nonlinear tube-based model predictive control
PSD	Positive semidefinite
RBF	Radial basis function
RNN	Recurrent neural network
SDP	Semidefinite program
SOC	State of charge

SOS . . . . . Sum of squares

TAS . . . . . True airspeed

TMPC . . . . Tube-based model predictive control

UAM . . . . . Urban air mobility

VTOL . . . . . Vertical take-off and landing

1 Introduction

Contents

1.1 Motivation

Control theory and optimisation are central to solutions for decarbonising the aviation industry and meeting the requirement of net zero emissions by 2050 [1, 2]. They provide a rigorous framework for optimal decision making in the context of green aviation and urban air mobility (UAM).

The main challenges raised by control and optimisation of aerospace systems are: i) the uncertainty and complexity of the environment in which vehicles operate (e.g. wind affecting aircraft trajectories); ii) the need for real-time global solutions where algorithms often involve nonconvex large-scale optimisation problems with multiple local minima; iii) the safety-critical nature of the sector that requires the massive deployment of provably safe algorithms with guarantees of convergence.

Robust optimisation and robust model predictive control (MPC) are well suited to address these challenges. In robust optimisation, the constraints and/or cost

functions are subject to uncertainty and a solution is sought that minimises the worst case cost while satisfying the constraints for all realisations of the uncertainty. Robust MPC is concerned with generating control sequences in order to guarantee safe operation in the presence of disturbances or model uncertainty. These approaches are promising in air transport scenarios where robustness and safety are vital requirements. However, a direct application of these methods to realistic problems requires the solution of numerically intractable optimisation. To construct reliable and efficient algorithms that can address these issues, the optimisation problems can be systematically reformulated as convex programs of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 0, \ldots, m \\ & h_j(x) = 0,\ j = 0, \ldots, n \end{aligned}$$

where $f,\ g_i : \mathbb{R}^n \to \mathbb{R}$, are convex functions of x and $h_j : \mathbb{R}^n \to \mathbb{R}$ are affine functions $\forall i, j$, and for which globally optimal solutions can be obtained with predictable computational effort.

Contrary to previous work in the literature, approaches based on convex robust optimisation and convex robust MPC offer new perspectives in terms of: i) optimality with respect to energy consumption; ii) robustness in the presence of uncertainties inherent to air transport scenarios; iii) rapid, global convergence. These advantages promise to allow large scale real-time implementations compliant with national and international safety certification requirements.

This thesis investigates the use of these approaches to solve emerging problems arising from green aviation and UAM:

- A first case study is concerned with energy management of hybrid-electric aircraft equipped with gas turbine and electric motor. The goal is to arbitrate in real time the power demand between the components of the propulsion system. A solution in terms of a systematic convex reformulation of the energy management optimisation and an MPC decision law is proposed.

- In a second case study, trajectory optimisation of a tiltwing vertical take-off and landing (VTOL) aircraft is investigated to achieve transitions from hover to forward flight in the context of UAM. A difference of convex functions (DC) decomposition of the dynamics is computed and convexity is exploited to infer tight bounds on the necessarily convex linearisation error of the dynamics, allowing to define a robust optimisation where these errors are treated as bounded disturbances.

- In a third case study, robust control of VTOL aircraft subject to wind disturbances is considered. The DC robust optimisation described above is leveraged in a robust MPC framework to develop a new control algorithm that achieves the objectives of optimality, robustness and computational tractability.

The unifying theme across all case studies is the use of convex optimisation to develop new algorithms that are computationally tractable and offer guarantees of convergence to a global optimum. Convexity is exploited with techniques such as MPC and robust optimisation to meet the stringent requirements set by aeronautic regulations in terms of robustness and optimality.

1.2 Outline

The main contributions of the thesis are organised into the following chapters:

- **Chapter 2.** We review the literature related to energy management of hybrid-electric aircraft, trajectory optimisation of tiltwing VTOL aircraft and robust MPC based on tubes. We identify gaps in the literature that justify the contributions made by this thesis and the research undertaken.

- **Chapter 3.** We consider the problem of energy management for a hybrid-electric aircraft. Through a convex formulation of mathematical models of the propulsion system, a computationally tractable optimisation problem is constructed whose globally optimal solution is used to arbitrate in real time

the power demand of the aircraft between the gas turbine and electric motor, allowing significant fuel savings. Computation times are also reduced by an order of magnitude thanks to a custom implementation of a first order solver, enabling fast real-time implementations for new generations of more-electric aircraft.

This chapter is based on the following publications:

- [3] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. "Optimal energy management for hybrid electric aircraft". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 6043–6049.
- [4] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. "Predictive energy management for hybrid electric aircraft propulsion systems". In: *IEEE Transactions on Control Systems Technology* 31.2 (2022), pp. 602–614.
- [5] Martin Doff-Sotta, Mark Cannon, and James Forbes. "Spacecraft energy management using convex optimisation". In: *Energy Conversion and Management: X* 16 (2022).

- **Chapter 4.** We investigate trajectory optimisation for a tiltwing VTOL aircraft to achieve transitions from hover to forward flight in an urban air mobility (UAM) scenario. The approach is based on successively linearising the system around guess trajectories and treating the linearisation error as a bounded disturbance in a robust optimisation framework. A novel method is developed by decomposing the dynamics as a difference of convex functions (DC) and exploiting convexity to obtain a computationally tractable optimisation with robustness guarantees to both exogenous disturbances and model-based uncertainty. Crucially, by convexity of the dynamics, the linearisation error is necessarily convex and takes its maximum at the boundary of the uncertainty set, allowing tight bounds to be computed. This provides a very general method for solving uncertain optimisation problems as a

sequence of computationally tractable convex programs where the dynamics are approximated tightly by a set of convex inequalities.

This chapter is based on the following publications:

- [6] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. "Fast optimal trajectory generation for a tiltwing VTOL aircraft with application to urban air mobility". In: *2022 American Control Conference (ACC)*. IEEE. 2022, pp. 4036–4041.
- [7] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. "Robust trajectory optimisation for transitions of tiltwing VTOL aircraft". In: *2023 Conference on Control Technology and Applications (CCTA)*. IEEE. 2023.

- **Chapter 5.** This chapter leverages the robust optimisation based on DC decomposition developed previously in a robust MPC framework. We first present a computationally tractable tube-based MPC algorithm for nonlinear systems representable as DC functions. The key idea is to exploit the convexity in the dynamics to bound system trajectories tightly and obtain the robust MPC law from a sequence of convex programs. The approach can then be generalised to any sufficiently regular nonlinear system through DC decomposition of the dynamics. In particular, it can be applied to safety-critical aerospace systems by providing a safe, robust and optimal control command with rapid, global convergence. We establish theoretical results such as recursive feasibility of the scheme, stability and convergence. We then propose extensions of the scheme to systems subject to additive disturbance and to systems with twice-continuously-differentiable dynamics.

  This chapter is based on the following publications:

  - [8] Martin Doff-Sotta and Mark Cannon. "Difference of convex functions in robust tube nonlinear MPC". In: *2022 Conference on Decision and Control (CDC)*. IEEE. 2022.

  - [9] Niels Krausch, Martin Doff-Sotta, Mark Canon, Peter Neubauer, and Mariano Nicolas Cruz Bournazou. "Handling nonlinearities and uncertainties of fed-batch cultivations with difference of convex functions tube MPC."
  - [10] Johannes Buerger, Mark Cannon and Martin Doff-Sotta. "Safe Learning in Nonlinear Model Predictive Control." In: *6th Annual Learning for Dynamics and Control Conference*. University of Oxford. 2024.

- **Chapter 6.** Building on the developments of the previous chapter, we present a computationally tractable data-driven approach to robust MPC of VTOL aircraft subject to wind disturbances in UAM scenarios. The resulting control scheme offers robustness guarantees to model uncertainty and exogenous disturbances, paving the way to future widespread adoption in safety-critical applications related to decarbonisation of the transport sector, green aviation and UAM.

  This chapter is based on the following publication (under review, conditionally accepted to the Journal of Guidance, Control and Dynamics):

  - [11] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Data-driven robust model predictive control of tiltwing vertical take-off and landing aircraft”. In: *arXiv preprint arXiv:2308.03557 (2023), arXiv–2308*, submitted to AIAA Journal of Guidance, Control and Dynamics.

2
Literature review

Contents

2.1 Introduction

We review the state of the art for the themes investigated in this thesis and identify knowledge gaps. The first application considered is energy management for hybrid-electric aircraft. We review the different types of hybrid-electric aircraft architectures and the energy management strategies employed. The second application is related to trajectory optimisation of vertical take-off and landing (VTOL) aircraft

in the context of urban air mobility (UAM). We consider the specific case of the tiltwing topology whose higher endurance is particularly desirable in UAM scenarios and we review trajectory optimisation of tiltwing VTOL aircraft. The third application is concerned with robust model predictive control (MPC) of VTOL aircraft. We contextualise briefly robust MPC and in particular consider computationally tractable implementations based on robust tubes. We give a detailed account of tube-based MPC in the context of linear and nonlinear systems.

2.2 Energy management for hybrid-electric aircraft

Aviation is currently responsible for producing around 2% of global CO2 emissions and represents 12% of the total emissions due to transportation [12]. In line with concerns about global warming, objectives have been set to reduce by 50% the footprint of the aerospace industry as part of FlightPath 2050 [13]. One avenue identified to achieve this ambitious goal is the development of a greener aviation based on new propulsion concepts, in particular the use of hybrid-electric drivetrains for short-haul flights and the operation thereof.

2.2.1 Hybrid-electric aircraft

Aircraft powered with hybrid-electric propulsion architectures were shown to achieve significant emission reduction and energy savings [14–16], this trend being concurrent with recent advances in battery technology and hydrogen-based propulsion systems [17]. In [14], a commercial airliner equipped with a turboelectric propulsion system is considered, and simulation showed that it could achieve fuel savings of up to 7% by contrast to conventional propulsion architectures. The system consists of two wing-mounted turbofans with generators powered by the fan shaft and producing electricity transferred to a boundary layer ingesting propulsor. A distributed electric propulsion concept for the transonic cruise range was proposed in [18] and is also expected to benefit from a 7% reduction in fuel. Similar savings and 1.5% NOx level reduction were shown for a parallel-hybrid Airbus A320 with

predicted technology from 2035 [15]. In [16], significant potential energy savings were demonstrated for a concept year-2030 aircraft equipped with a parallel-hybrid propulsion system combined with an all-electric propulsion system. The benefits of an electric propulsion system on the aerodynamic performance of a four-seat aircraft was shown in [19] where the additional lift generated by the slipstream on the wing from electrically-powered propellers yields a drastic increase of efficiency. In [20], a parallel-hybrid configuration where the battery is used to absorb peak loads is shown to achieve a potential 4% decrease in fuel by comparison with benchmark gas-turbine-powered aircraft. A regional series-hybrid aircraft, the E-fan X [17], was under development by Airbus and Rolls-Royce before the project was prematurely cancelled in 2020.

Hybrid aircraft relying on hydrogen fuel have also been proposed in order to meet targets for reducing greenhouse gas emissions [17]. Hydrogen can be used to generate electricity in a fuel cell or for internal combustion instead of kerosene (liquid hydrogen being mixed with oxygen in a gas turbine). In the context of more-electric aircraft (i.e aircraft for which the traditionally-nonelectric subsystems have been replaced by new electrical subsystems [21]), there are growing incentives to replace the ram air turbine emergency power systems by hybrid systems relying on fuel cells, batteries and supercapacitors [22]. Recently, Airbus introduced the ZEROe program with the ambition to develop a zero-emission hydrogen-powered commercial aircraft by 2035. The hybrid propulsion system consists of a liquid-hydrogen-powered gas turbine complemented by hydrogen fuel cells producing electricity [23].

Two main types of hybrid propulsion architectures are considered in the literature: parallel-hybrid and series-hybrid configurations [24]. In the parallel architecture, a gas turbine is mechanically coupled with an electric motor in a parallel arrangement, and the common shaft drives the fan or propellers. In the series architecture, the net power output of the propulsion system is delivered by an electric motor which takes its electrical power from two sources: a battery and a turbo generator set (gas turbine in series with an electric generator). Other variations of these basic arrangements exist, such as the series-parallel configuration.

2.2.2 Energy management strategies

Hybrid-electric propulsion systems require energy management strategies in order to allocate the power demand between the different components of the drivetrain (gas turbine and electric motor). Energy management is as important as component sizing when considering the problem of minimising fuel consumption of hybrid vehicles. Simple strategies were proposed, for example, the charge-depleting-charge-sustaining (CDCS) policy which consists in using the battery until depleted and then relying on the engine for the rest of the mission [25]. A peak shaving strategy was proposed in [20] where the battery is used whenever the gas turbine reaches its maximum power output. A solar-powered high-altitude long-endurance aircraft is considered in [26], where the excess solar energy is stored in the form of gravitational potential during daytime and released at night by gliding. Approaches that use state machines [27] are based on monitoring critical variables (e.g. the state of charge of the battery, the load demand, etc.) and decisions are taken accordingly in a rule-based "`if - then`" fashion. These algorithms can be fine-tuned as experimental data accumulate and expertise on the system increases [22]. Rule-based fuzzy logic techniques [28] use a state machine augmented with (trapezoidal) membership functions for the decision variables, which increases robustness with respect to measurement errors. The transition parameters of the membership functions can be tuned using e.g. genetic algorithms. In [29], an operation policy for the battery is proposed that finds a trade-off between two extreme strategies (namely peak shaving and maximum battery utilisation). Some approaches seek to minimise aging of the components of the hybrid system. For example, the frequency decoupling strategy [30] consists in assigning the slowly-varying low-frequency content of the load profile to critical components (e.g. fuel cells that age rapidly when submitted to dynamic loadings) while high-frequency variations can be delivered by components capable of absorbing dynamic stresses (e.g. supercapacitors). Frequency decoupling can be achieved using low pass filters, wavelet transforms, fast Fourier transforms [22] or using a model predictive control strategy [31]. Approaches based on neural network [32] and neuro-fuzzy adaptive control [33] were also proposed.

Optimisation techniques that seek to minimise a cost function (e.g. fuel consumption) are also popular to solve the energy management problem. For example, the so-called equivalent fuel consumption minimisation strategy is widely used in hybrid fuel cell systems [34]. The optimal power split is obtained by constrained minimisation of the fuel consumption of the cell and the equivalent fuel consumption of the other energy sources, while meeting the load demand and keeping all energy sources in their allowed operating range. It is worthwhile mentioning that this strategy considers simple static models of the energy sources, and could thus be extended with more comprehensive dynamical models. A different approach is proposed in [35] for the emergency power system of an aircraft: instead of estimating the equivalent fuel consumption of electric energy sources, minimisation of hydrogen consumption is achieved by maximising consumption of stored electrical energy. By contrast to methods based on equivalent fuel consumption, using the battery and supercapacitor power consumption directly in the cost function simplifies greatly the approach and increases robustness to changes in the load profile. Dynamic programming techniques [36–38] search for the global-optimal policy. Such methods can be used offline to provide a comparison reference, but their computation times are prohibitive for a real-time implementation. Other approaches based on $\mathcal{H}_\infty$ control [39] and optimal adaptive control [40] were also reported. A popular framework for the energy management problem is model predictive control (MPC), a topic that has been widely explored for electric and hybrid-electric ground vehicles [41–43]. The energy management problem is formulated as a receding-horizon constrained optimisation problem, and an optimal power split is found at each time step. MPC offers a degree of robustness to modeling uncertainty and prediction errors, along with updated feedback information on the state at each time step.

MPC-based energy management involves solving a constrained optimisation problem in real time. Various optimisation techniques were proposed to address that problem [44], such as interior point [42, 45, 46] and active set [47] methods. Recently, there has been a growing interest for first-order methods, whose computational efficiency makes them highly desirable candidates for online energy management.

For example, fast gradient methods [48, 49] or the alternating direction method of multipliers (ADMM) [50] are powerful techniques suitable for a real-time implementation. ADMM-based solvers exploit the separability of the constraint and cost functions to update the optimisation variables sequentially. The ADMM algorithm was applied to the energy management problem for hybrid-electric ground vehicles in [42, 51]. An online receding-horizon optimisation problem, including nonlinear dynamical constraints, was formulated as a convex program and an ADMM solver was shown to outperform other optimisation schemes in terms of computation time while retaining an acceptable precision.

2.2.3 Identified gap

Although MPC was recently applied to solve the energy management problem for hybrid-electric aircraft [52, 53], none of these approaches considered lossless convexification of the nonlinear problem, a strategy that proved its efficiency in electric ground vehicles. Such an use case of convex optimisation in energy management of hybrid-electric aircraft would be computationally efficient, offer guarantees of convergence to a global optimum, and could be leveraged in multi-stage optimisation schemes for a real-time update as new data are collected.

2.3 Trajectory optimisation of VTOL aircraft

The increase of congestion and pollution levels in large metropolitan areas has recently incentivised the development of sustainable and green mobility solutions. As an example, in 2014 in the USA, road traffic has generated more than 11 million litres of excess fuel and nearly 7 billion person-hours were wasted - the equivalent of about 10,000 people average lifespans [54]. Urban air mobility (UAM) has the potential to address these problems by allowing air transport of goods and people [55], thus reducing the pressure on urban traffic. It has been estimated that 160,000 air taxis could be in circulation worldwide by 2050, representing a USD 90 billion market [56]. A key enabler of UAM is a class of zero carbon emission eVTOL aircraft concepts [57], many based on tiltrotor, tiltduct or tiltwing vehicle configurations

powered by batteries or by making use of zero carbon fuel like hydrogen [58]. However, the lower energy density of energy storage options relative to liquid carbon based fuels restrict the operational range, hover time and cruise speeds.

2.3.1 Tiltwing VTOL aircraft

Tiltwing vertical take-off and landing (VTOL) aircraft, with their capability to take-off and land in restricted spaces and their extended endurance, have recently received a lot of attention in a UAM context. Despite being investigated since the 1950's [59–61], issues related to control and stability, as well as the mechanical complexity associated with the use of conventional gas turbine engines have prevented widespread adoption of this technology. However, recent advances in battery and electric motor technology, combined with the development of modern control systems architectures, has spurred a new interest for these vehicles [62]. Some recent prototypes based on tiltwing configurations are the Airbus A^3 Vahana [63], the NASA GL-10 [64], or Rolls-Royce's eVTOL aircraft [65].

The transition manoeuvre between vertical and horizontal flight for a VTOL aircraft is the most critical phase of flight. Consequently, limited energy and transition envelope constraints impede the throughput of flight operations by limiting the number of flights, timely allocation of landing slots, duration of holding patterns in hover, and separation between vehicles. To maximise throughput it is clear that both the energy spent in a non-wing-borne flight phase and the time and space needed to transition between thrust-borne and wing-borne flight should be minimised.

2.3.2 Trajectory generation for VTOL aircraft

Although heuristic strategies have been proposed based on smooth scheduling functions of the forward velocity or tilt angle [66], trajectory optimisation for the transition of VTOL aircraft is still an open problem as it involves determining the optimal combination of thrust and tiltwing angle to minimise an objective while meeting constraints on system states and control inputs, which requires solving large-scale nonlinear optimisation problems.

The problem of determining minimum energy speed profiles for the forward transition manoeuvre of the Airbus A^3 Vahana tiltwing VTOL aircraft was addressed in [67]. The vehicle is modelled in all three phases of flight considered: cruise, transition and descent. The energy consumption in all flight phases is established and a flight profile is searched that minimises the total energy while satisfying the dynamic and kinematic constraints. A major drawback of the approach is that the tiltwing angle was not included in the optimisation and the transition was assumed to occur instantly, which is unrealistic for such a vehicle. In [62], the trajectory generation problem for take-off is formulated as a constrained optimisation problem and solved using NASA's OpenMDAO framework and the SNOPT gradient-based optimiser. A case study based on the Airbus A^3 Vahana is considered, including aerodynamic models of the wing beyond stall, and the effect of propeller-wing interaction. The drive power is minimised considering the wing angle, electric power and flight time as optimisation variables with constraints on final altitude and terminal cruise speed. Additional constraints on angle of attack, acceleration and horizontal distance are also considered. A 5% energy saving is achieved using the proposed optimisation when compared to a baseline situation where the aircraft transitions at a constant optimal climb rate. It is shown that the optimal trajectory involves operating the wing in near-stall conditions, although the benefit of doing so is practically negligible. Forward and backward optimal transition manoeuvres at constant altitude are computed in [68] for a tiltwing aircraft, considering leading-edge fluid injection active flow control and the use of a high-drag device. The effect of propeller vortex interaction with the wing is modelled and included in the model of the vehicle. The optimisation problem is transcribed into a nonlinear program (NLP) using GPOPS-II, a general-purpose nonlinear optimisation software implementing a pseudospectral collocation method at the Legendre-Gauss-Radau points. The NLP is then solved using the IPOPT solver implementing a primal-dual interior point method.

2.3.3 Identified gap

All of these approaches consider optimising trajectories using general purpose NLP solvers without exploiting potentially useful structures and simplifications. These approaches are computationally expensive and not generally suitable for real-time implementation.

Convexification of the optimal trajectory generation problem based on changes of variables, lossless reformulations, mild simplifications, and the use of decomposition methods has the potential to yield much faster, reliable solutions and could be leveraged in real time, e.g. to reschedule the trajectory in the event of an obstacle.

2.4 Robust MPC for nonlinear uncertain systems

The development of provably safe autopilots is a prerequisite for the deployment of tiltwing VTOL aircraft in non-segregated airspace within the context of UAM. The main difficulty arises from the presence of uncertainties in the system, in the form of unmodeled dynamics, parametric uncertainties or unknown external disturbances. Moreover, in the context of minimising the footprint of transportation on climate change, it is also desirable for such vehicles to achieve good performance in terms of energy consumption.

2.4.1 Robust MPC

Robust MPC methods form a family of controllers which are particularly well suited for energy-efficient and reliable operation of systems subject to uncertainty. Their deployment in industrial applications has recently been made possible thanks to the improvement of computational capabilities of microcontrollers and also as a result of the development of algorithms that use simplified optimisation strategies (e.g. robust MPC relying on the concept of tubes [69] to parameterise a controller and to bound the closed-loop trajectories). Robust MPC is concerned with preserving performance and closed-loop stability for systems subject to uncertainty [69], while offering the properties of optimality, real-time tractability and constraint satisfaction

of classical MPC. These methods are based on the idea that the knowledge of a set bounding the uncertainty in the dynamics allows one to define a sequence of reachable sets where the system future inputs and states are guaranteed to remain in [69]. Uncertainty may be present in a system under various forms [70]. Additive uncertainty typically appears in systems subject to external disturbances

$$x_{k+1} = f(x_k, u_k) + w_k,$$

where $x_k \in \mathcal{X}$ is the state, $u_k \in \mathcal{U}$ is the control input, and $w_k \in \mathcal{W}$ is the disturbance. Parametric uncertainty arises from lack of knowledge or unknown variations in the parameters of the model and can be modeled in general by

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

where $\theta_k \in \Theta$ represents the uncertain parameters of the system. In the linear case, this type of uncertainty appears in the form of a linear-time-varying (LTV) system with so-called multiplicative (or polytopic) uncertainty [71]

$$x_{k+1} = A_k x_k + B_k u_k,$$

where $\theta_k = (A_k, B_k)$ is assumed to belong to a compact convex polytopic set constructed as the convex hull of a known set of vertices $\Theta = \mathrm{Co}\{(A^{(i)}, B^{(i)}) \,|\, i = 1, \ldots, m\}$, such that $(A_k, B_k) = \sum_{i=1}^{m} \lambda_i (A^{(i)}, B^{(i)})$ for some positive λ_i such that $\sum_i \lambda_i = 1$. Systems with mixed uncertainty, i.e. combining both types of uncertainty, can also occur.

In robust control approaches, uncertainties are usually assumed to lie in a compact set, whereas stochastic approaches rely on a description of disturbances as random processes, i.e. sequences of random variables following probability distributions and taking values in sets that are not necessarily compact [72]. We focus here on robust approaches only, see [73] for a complete review of stochastic MPC.

The concept of designing robust-optimal controllers for constrained systems subject to unknown disturbances dates back from the end of the 1960s [74–76]. These techniques were all based on the minimisation of a cost function computed assuming

the worst case realisation of the uncertainty over the horizon. Such conservative approaches originate from game theory and are referred to as min-max optimisation problems because they involve the concurrent maximisation of the cost to account for the worst-case disturbances and minimisation of the resulting worst-case cost to generate an optimal control sequence $u^* = \arg\min_{u_k} \max_{w_k} J(x_k, u_k, w_k)$ subject to constraints. The problem of reachability for uncertain systems, i.e finding a control law that drives the system subject to disturbances into some target set, is also considered in these papers.

An early robust MPC framework was formulated in [77] for stable uncertain linear-time-invariant (LTI) systems. A worst-case objective is minimised subject to inputs and states constraints, and the problem of finding the optimal input sequence reduces to a finite-dimensional linear program.

Another attempt of robust MPC applied to linear systems subject to additive uncertainty is described in [78] where robustness is guaranteed by tightening of the constraint sets of the nominal system, and conditions for recursive feasibility are provided. Tightening of the constraint sets is usually achieved offline by solving a finite series of linear programs in order to determine a conservative upper bound on the size of the uncertainty sets (outer approximation) for the system inputs and states along a predicted trajectory.

In [71], the robust MPC problem for LTV systems with model uncertainty is treated with a min-max approach and use of linear matrix inequalities (LMI). A linear state feedback control law is efficiently computed at each time step that minimises a worst-case upper bound on the performance objective while satisfying inputs and states constraints. The uncertain system trajectories are shown to stay within invariant ellipsoidal sets at each time step and robust asymptotic stability of the closed loop is demonstrated. The results apply also to systems with additive disturbances. Constraints on the worst-case realisation of the states and inputs of the uncertain system are enforced via an additional set of LMIs included in the MPC problem. The approach suffers from a significant computation burden since the feedback matrix has to be recomputed at each time step and the

invariant sets are limited in size. It also suffers from the potentially conservative assumption that the linear feedback law, although optimised, is constant over the future prediction horizon.

Later approaches applied to linear systems and involving a static state feedback control law augmented with a feedforward term computed online [79–82] greatly improve these aspects. These techniques permit computationally tractable implementations [69] by contrast to searching for arbitrary admissible nonlinear policies and form the basis of MPC algorithms using tubes.

2.4.2 Tube-based MPC of linear systems

Tube-based MPC (TMPC) is a specialised robust MPC technique that conveniently relies on a parameterisation of the uncertainty sets to construct a tube in which the system trajectories are guaranteed to remain at all future instants and for all realisations of the uncertainty [69]. A tube can be viewed as a collection of reachable sets (the tube "cross-sections") centred[1] around the nominal system trajectory (the tube "central line").

Such an approach works particularly well in the case of uncertain linear systems of the form

$$x_{k+1} = Ax_k + Bu_k + w_k.$$

The idea is that – the system being linear – it is possible to isolate the contribution of the disturbance on the system by independently considering the nominal dynamics as

$$z_{k+1} = Az_k + Bv_k,$$

where $z_k \in \mathcal{Z}$ is the nominal (undisturbed) state, $v_k \in \mathcal{V}$ is the nominal control input. Defining the deviation of the actual state from the nominal state as $e_k = x_k - z_k$, we can design a two-tier controller for the uncertain system as follows

$$u_k = v_k + Ke_k,$$

[1] Although it does not always have to be a geometrical centre of the set, see e.g. Chebyshev centres.

where the feedforward term v_k is stabilising the nominal system and the state feedback term Ke_k is separately responsible for regulating the disturbances. Such a controller allows one to control the deviation dynamics

$$e_{k+1} = \Phi e_k + w_k, \quad \Phi = A + BK.$$

This can be reformulated in terms of the dynamics of uncertainty sets $e_k \in \mathcal{S}_k$

$$\mathcal{S}_{k+1} \supseteq \Phi \mathcal{S}_k \oplus \mathcal{W},$$

for which a closed form solution exists and is given by $\mathcal{S}_k = \bigoplus_{i=0}^{k-1} \Phi^i \mathcal{W}$, assuming that the deviation is initially zero ($\mathcal{S}_0 = \{0\}$). This offers a unique way of controlling the evolution of the cross-sections of the tube defined by $\{z_k\} \oplus \mathcal{S}_k$, assuming a known compact set $\mathcal{W}$ for the disturbances and that the matrix Φ can be made stable (via proper design of K). Note that this approach can be generalised to LTV systems with time varying A_k, B_k, K_k matrices.

To achieve robust stabilisation of the system into a tube, the two-degrees-of-freedom controller is implemented in two stages:

- The feedforward control sequence $v = \{v_k,\, k = 0, 1, \ldots\}$ is computed online by solving an MPC problem for the nominal system with tightened state, input and terminal constraint sets $(\mathcal{Z}, \mathcal{V}, \mathcal{Z}_N) = (\mathcal{X} \ominus \mathcal{S}, \mathcal{U} \ominus K\mathcal{S}, \mathcal{X}_N \ominus \mathcal{S})$ accounting for the presence of uncertainty, where $\mathcal{S}$ is an outer approximation for the tube. At each time step k, the following optimisation problem with horizon N is solved

$$\begin{aligned} \min_{v} \quad & ||z_N||^2_{Q_N} + \sum_{i=0}^{N-1} (||z_i||^2_Q + ||v_i||^2_R) \\ \text{s.t.} \quad & z_{i+1} = Az_i + Bv_i,\ \forall i = 0, \ldots, N-1, \\ & z_i \in \mathcal{X} \ominus \mathcal{S},\ \forall i = 0, \ldots, N-1, \\ & v_i \in \mathcal{U} \ominus K\mathcal{S},\ \forall i = 0, \ldots, N-1, \\ & z_N \in \mathcal{X}_N \ominus \mathcal{S}. \end{aligned} \tag{2.1}$$

  and the first element of the optimal control sequence v^* is retained by the controller to generate the central line of the tube: $v(kT) \leftarrow v_0^*$. At the next

time step, $k \leftarrow k+1$, the problem is updated with $z_0 \leftarrow x(kT)$ (assuming $x_0 = z_0$) and the process is repeated in a receding horizon fashion.

- The size of the tube cross-sections is restricted via a state feedback controller that steers the uncertain system trajectories around the central path. For linear systems, a feedback matrix is usually computed offline, e.g. as the solution of a linear quadratic regulator (LQR) problem associated to the nominal unconstrained system, an approximation that is still optimal for the constrained system in a neighbourhood of the nominal trajectory [70]. Other robust techniques might be used such as those relying on $\mathcal{H}_\infty$ or mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control [81]. Alternatively, the feedback matrix can also be optimised online subject to LMI constraints [80] by solving a sequence of semidefinite programs [83]. The motivation for an online solution comes from the fact that in the presence of uncertainties, the feedback matrix is strongly state dependent, and performance is thus improved greatly when the feedback matrix is recomputed at each time step vs. using a static gain computed offline [71]. Online optimisation of the stabilising feedback controller might however increase computation time significantly [84].

The idea of using tubes in control is not new, see [76, 85, 86]. In [76], algorithms for the offline computation of ellipsoidal and polytopic tubes are proposed and the control law is obtained via a min-max approach. Precursor approaches to TMPC were introduced in [78–80], without explicitely coining the term "tube" nor making a geometrical interpretation in terms of tubes. For example, the idea of set tightening for linear systems subject to additive disturbances is detailed in [78].

In [79], a control law consisting of a feedforward and state feedback term is proposed. The feedforward term is computed by minimisation of a nominal cost subject to constraint sets with online degrees of freedom in their shape (parameterised as low complexity polyhedra). The optimisation problem is formulated as a convex program subject to LMI constraints that can be solved efficiently with semidefinite programming.

Instead of tightening the constraint sets, a different strategy based on finding a tight upper bound on the nominal cost for a LTV uncertain system is proposed in [80]. The robustly stabilising control law consisting of a feedforward and state feedback term is obtained by minimising the upper bound of the cost - which is itself a variable of the proposed constrained LMI optimisation problem. Using the objective upper bound as an optimisation variable yields a less conservative solution than with min-max strategies where the worst case upper bound is chosen, but this extra degree of freedom in the optimisation can be computationally expansive.

In [87], a robust predictive controller that steers the system state in a minimal robust invariant set is investigated[2]. This is achieved by construction of a two-degree-of-freedom controller with a state feedback term designed to provide optimal performance and a feedforward term obtained by minimisation of a cost for the nominal system subject to tightened constraints. The controller guarantees feasibility for all future states under all admissible disturbances realisations. The approach is restricted to time-invariant feedback matrices that have to be computed offline.

A controller is proposed in [88] for LTI or LTV systems with model uncertainty and / or additive disturbances. The solution of the optimal control problem is a sequence of sets forming a tube that contains the system trajectories and the associated time-varying control law. The proposed control law has the form of a convex combination of control laws acting on the vertices of the sets defining the tube, so that the system is guaranteed to stay inside the tube at all time. Although applicable to problems with multiplicative uncertainty in principle, it does not work well in this context because the state tubes are based on minimal robust invariant sets, which are difficult to compute for LTV systems.

Systems subject to multiplicative uncertainty are more challenging to deal with due to the exponential increase in complexity: for a polytopic uncertainty set with ρ vertices, the trajectory of the system N steps ahead lies in a polytope with ρ^N vertices, making the approach intractable for large dimensions or large

[2]A set Ω is robust positive invariant for the system $x_{k+1} = f(x_k, w_k)$ and constraint sets $(\mathcal{X}, \mathcal{W})$ if $\Omega \subseteq \mathcal{X}$ and $f(x_k, w_k) \in \Omega$, $\forall w_k \in \mathcal{W}$, $\forall x_k \in \Omega$. The minimal robust invariant set Ω^* is such that every other invariant set Ω satisfies $\Omega \supseteq \Omega^*$ [70].

prediction horizons [89]. A different strategy was proposed in [70] where an additive uncertainty is introduced in the nominal dynamics to bound the multiplicative uncertainty as follows

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k + w_k,$$
$$w_k = (A - \bar{A})x_k + (B - \bar{B})u_k,$$

where $(A, B) = \sum_{i=1}^{m} \lambda_i (A^{(i)}, B^{(i)})$, $\sum_{i=1}^{m} \lambda_i = 1$, $\lambda_i \geq 0$, $\forall i$, and $\bar{A} = 1/m \sum_{i=1}^{m} A^{(i)}$, $\bar{B} = 1/m \sum_{i=1}^{m} B^{(i)}$. This approach however could be overly conservative as it relies on a disturbance set that now depends on the state and input. A computationally tractable and less conservative alternative is proposed in [90] where the multiplicative uncertainty is treated directly.

Although most of the approaches for TMPC rely on the more or less arbitrary offline computation of the stabilising feedback matrix, a different approach is proposed in [84]. The control policy is still parameterised by a two-degree-of-freedom affine control law involving a state feedback stabilising term and an optimal control sequence, but the state feedback matrix is now part of the optimisation and the previous states are used in the feedback law. Equivalence between the affine state feedback law with memory of prior states parameterisation and the affine parameterisation in terms of past disturbances is shown, which makes it possible to define the constrained receding-horizon optimal problem for a linear system subject to additive disturbances as a computationally tractable convex program. The closed loop is shown to be input-to-state stable[3] (ISS) and the method offers a less conservative solution than other approaches that do not optimise the state

[3]Roughly speaking, a system is ISS if: i) its state trajectory is bounded for a bounded input; and ii) its trajectory eventually becomes small if the input becomes small, regardless of the initial condition [91]. More rigorously, the uncertain system $x_{k+1} = f(x_k, \kappa(x_k), w_k)$ under control law $\kappa(\cdot)$ and subject to a sequence of external disturbances $w = \{w_k,\, k = 0, 1, \ldots\}$ is ISS if there exists $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$ such that $\forall x_0$, $\forall k$

$$||\phi(k, x_0, w)|| \leq \beta(||x_0||, k) + \gamma(\{\sup ||w(\tau)|| \,\text{s.t.}\, \tau \in \mathbb{Z}_{[0,k-1]}\}),$$

where $||\cdot||$ denotes the Euclidean norm, $\phi(k, x_0, w)$ is the trajectory of the system at k starting from initial condition x_0 and under the disturbance sequence w. We denote by $\mathcal{K}$ the set of continuous, strictly increasing functions γ with $\gamma(0) = 0$. We denote by $\mathcal{KL}$ the set of functions $\beta(s, k)$ such that $\forall k \geq 0$, $\beta(\cdot, k) \in \mathcal{K}$ and $\forall s \geq 0$, $\beta(s, \cdot)$ is decreasing with $\lim_{k\to\infty}, \beta(s, k) = 0$. See [91] for a more complete treatment of ISS.

feedback controller online. Although tractable at low problem dimensions, the approach presented in [84] is computationally costly, and further improvements based on a reformulation into a structured and sparse problem have been introduced in [92] that reduce the computational complexity from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^3)$ operations at each time step with interior-point methods, where N is the horizon length.

A modern description of TMPC applied to linear systems can be found in [93] where the usual two-degree-of-freedom controller is used with tightened sets and the initial state of the system is taken as one of the optimisation variables. The controller guarantees convergence to robustly exponentially stable sets for the closed loop subject to disturbances.

Several tube parameterisations have been proposed for TMPC. Algorithms with rigid tubes [93] assume that the sets characterising the uncertainty distribution are of constant size, considering the steady state effects of disturbances. This might be conservative given that it takes some time for the uncertainty in the state to build up to steady state levels. Homothetic tubes [94] are proposed in order to alleviate this problem, where the size of the sets are scaled by a sequence of parameters that are optimised online. These approaches are superior to those using rigid tubes as they allow to formulate less conservative bounds on the effect of disturbances on the future state during transients. However, homothetic tubes involve a larger optimisation problem for the feedforward controller, including constraints on the scaling parameters and modification of the nominal MPC cost. Polytopic tubes [79, 95] - although similar in spirit to homothetic tubes - involve tube cross-sections parameterised as polytopes rather than just scaled versions of a given set, allowing for a greater flexibility in the tube shape and less conservativeness. By contrast to high dimensional polyhedrons, ellipsoids are more conservative but these simple geometric shapes involve fewer optimisation variables, which makes them suitable for online optimisation of the shape of the sets. Ellipsoidal tubes are used for example in [83] where the size of the tube is optimised online.

2.4.3 Nonlinear tube-based MPC

In the context of robust optimal trajectory tracking of VTOL aircraft, which typically involve nonlinear dynamic equations subject to disturbances, it would be interesting to extend the theory of robust MPC to nonlinear uncertain systems of the form

$$x_{k+1} = f(x_k, u_k) + w_k.$$

The computational complexity involved to implement nonlinear MPC (NMPC) strategies still represent a barrier in their generalisation to real-life applications. Early attempts to robust NMPC (e.g. [96, 97]) yield formidably complex optimisation problems, which prevents tractable implementations. Indeed, application of robust MPC to nonlinear systems poses a series of challenges: online solution of nonconvex optimisation problems, exact description of uncertainty sets for states described by nonlinear dynamic equations, complexity associated with searching for a general nonlinear control policy, to name but a few.

Naturally, following the success of TMPC for linear systems, it is tempting to seek a solution to the robust NMPC problem using the concept of tubes. To achieve nonlinear TMPC (NTMPC), one would normally try to control the error dynamics between the uncertain state x_k and nominal state z_k given by

$$e_{k+1} = f(x_k, u_k) - f(z_k, v_k) + w_k.$$

While for linear systems the feedback law in the two-degree-of-freedom controller is usually linear, a nonlinear feedback policy $u_k = v_k + \kappa(x_k, z_k)$ should ideally be used in this framework [70]. The functions f and κ being nonlinear, the problem of designing a nonlinear feedback controller to control the evolution of e_k is *a priori* intractable. Moreover, the nominal control law v_k has to be obtained by solving a nonlinear MPC problem online, further complicating the task.

A NTMPC controller is proposed in [98] and [99] that computes the control law using two successive MPC controllers: i) on the one hand, the central line of the tube is obtained by minimising a nominal cost subject to nonlinear dynamics and tightened constraint sets (based on the idea of constraint tightening for nonlinear

system as introduced in [100]); ii) on the other hand, an ancillary MPC law is used to reject the disturbances acting on the system and maintain the trajectory within sets of known size centred along the tube line. The ancillary control law is obtained by minimisation of the error between the actual trajectory and the nominal trajectory computed with the first MPC controller. The computation of the tightened constraint sets is more difficult in the nonlinear case because the system describing the evolution of the error is no longer autonomous and the error is thus less easily bounded [70]. A tractable method for computing these bounds is proposed in [99] using offline Monte-Carlo simulations of the trajectories for various disturbances. Following an initial guess for the sets, tightening is refined based on the spread of the simulated trajectories. Also, [99] proposes the inclusion of feedback in the computation of the nominal MPC law by comparison between the cost function evaluated at the current state and nominal state, which proves superior to purely open-loop approaches.

A common strategy in NTMPC is to treat the nonlinearity in the dynamics as a bounded disturbance [83, 101–103]. For example, in [83], the proposed NTMPC algorithm is based on successive linearisations of the dynamics around predicted trajectories and uses a bound on the linearisation error to construct a tube and associated controller. At a given iteration, the dynamics are linearised around a guess trajectory (x^0, u^0) as follows

$$x_{k+1} = f(x_k^\circ, u_k^\circ) + A_k(x_k - x_k^\circ) + B_k(u_k - u_k^\circ) + g_k(x_k, u_k, x_k^\circ, u_k^\circ),$$

where $g_k \in \mathcal{G}$ is a bounded additive disturbance accounting for the Jacobian linerisation error. This system can thus be treated in the same tube-based formalism as for linear systems, using a two-tier parameterisation of the control input $u_k = v_k + K_k(x_k - x_k^\circ)$. After each iteration of the algorithm, the guess trajectories are updated with the solution of the MPC problem and the linearisation process is repeated successively until convergence. The controller achieves stabilisation of the nonlinear system within a tube formed with ellipsoidal sets with time-varying sizes, treating the linearisation error as a disturbance. Asymptotic stability under

the control law is demonstrated. The size of the sets and the feedback matrix are optimised online and the benefit is shown by comparison with a TMPC algorithm with fixed-size sets and offline-computed feedback matrix. The optimisation can be terminated early in order to reduce computational load without compromising closed loop asymptotic stability. However, the strategy has the disadvantage that convergence to a minimum point is not guaranteed, due to the use of linear bounds on linearisation errors.

While online optimisation of the constraint sets was addressed extensively for linear TMPC, e.g. [94, 104], very few works consider this problem applied to nonlinear systems. Although [83] investigates online optimisation of the tube for NTMPC, the approach is based on linearising the system around the nominal trajectory. The lack of results in tube optimisation for nonlinear system is due to the complexity of optimisation problems involving nonlinear relations between tube geometry and control parameters. The problem is addressed in [105], where the benefits of boundary layer sliding control are leveraged in order to simultaneously optimise the tube geometry and open-loop trajectories in real time. Indeed, the sliding mode control framework allows one to describe the tube shape dynamics by a first-order differential equation depending on the controller bandwidth and the uncertainty bound. One can thus optimise the tube geometry and feedforward control sequences through the control bandwidth parameter. This is achieved by augmenting the optimisation problem with the tube shape dynamics, and optimising for the control bandwidth that characterises both the controller and the tube geometry, resulting in less conservative tubes. The proposed method also includes a description of the uncertainty sets as a function of the state of the system, which, combined with the adaptive geometry capabilities of the tube, is used to achieve robust collision avoidance in simulation. The term "Dynamic tube-based MPC" is coined. Simulation results showed that, as expected, the tube size shrinks when in the vicinity of obstacles (which corresponds to a higher control bandwidth) and expands when in less cluttered environments (lower control bandwidth). Also, the

state dependence of the uncertainty bound is exploited by the algorithm, the velocity of the closed-loop being reduced by the controller when approaching an obstacle.

Other approaches involving the use of a sliding mode controller in a robust NMPC framework were proposed in [106, 107].

Sometimes, the special structure of the dynamics can be exploited to good advantage to define a NTMPC framework. An early example of tube-based receding-horizon control applied to nonlinear systems with a special structure is presented in [108]. The proposed controller takes the form of a feedback linearisation prewrap that cancels nonlinearities, followed by the usual linear control law that stabilises the system trajectory in a tube formed by robustly invariant sets. The method is applied to i) nonlinear systems with input-matched nonlinearities

$$x_{k+1} = Ax_k + B(g(x_k)u_k + \varphi(x_k)) + w_k,$$

for which the controller takes the form $u = g(x_k)^{-1}(-\varphi(x_k)+v_k+Ke_k)$ for invertible $g(\cdot)$, and ii) nonlinear processes that take the form of piecewise affine maps

$$x_{k+1} = A_j x_k + B_j u_k + c_j + w_k, \quad \forall x_k \in R_j,$$

where, in the special case $A_j = A + BC_j$, $B_j = BM_j$, $c_j = Bd_j$, the controller is defined by $u = M_j^{-1}(-d_j - C_j x_k + v_k + Ke_k), \forall x_k \in R_j$. This limits considerably the applicability of the method.

A method is proposed in [109] to construct robust invariant sets for a class of Lipschitz nonlinear systems, relying on the computation of a quadratic Lyapunov function and associated state feedback controller. ISS and ultimate boundedness are guaranteed for the closed-loop, along with recursive feasibility if the online optimisation problem has a feasible solution at the initial time. In practice, such approach is limited since finding a Lipschitz constant might be difficult.

A similar controller exploiting the existence of a Lipschitz constant is applied to ground vehicles in [110] in order to compute a robust invariant tube. The idea is to bound the nonlinear terms in the error dynamics assuming the Lipschitz structure. The error dynamics between the uncertain state x_k and nominal state z_k is given by

$$e_{k+1} = \Phi e_k + f(x_k) - f(z_k) + w_k, \quad \Phi = A + BK,$$

where $e_k = x_k - z_k$ and f is a nonlinear Lipschitz continuous function. This implies that $\exists L \geq 0$ such that

$$||f(x_k) - f(z_k)|| \leq L||x_k - z_k||, \quad x_k, z_k \in \mathcal{X},$$

where the smallest L' satisfying the inequality is the Lipschitz constant. This permits to naturally incorporate the nonlinearity as part of the additive disturbance and reformulate the error dynamics as a linear update equation with an augmented disturbance set $\tilde{w}_k \in \tilde{\mathcal{W}}$ as follows

$$e_{k+1} = \Phi e_k + \tilde{w}_k.$$

A linear quadratic feedback controller and associated tube are designed offline. The constructed tube is then used to tighten the constraint sets in the nominal MPC problem used to generate a collision free trajectory and open-loop input sequence for the vehicle. The usual two-degrees-of-freedom controller is then implemented and tested on a real passenger car driving on a snowy and icy terrain.

Nonlinear system dynamics with a differential flatness property can also be advantageously reformulated to define a tube-based receding-horizon controller as in [111].

Robustness of NTMPC is considered in the framework of ISS for nonlinear uncertain systems in [112]. Under mild assumptions on the cost function and constraint sets, the closed loop is shown to be ISS under a general nonlinear tube-based state feedback control law parameterised by the solution at each time step of the MPC optimisation for the constrained nominal system. ISS then guarantees global asymptotic stability of the nominal system and the existence of an asymptotic gain and stability margin for the disturbed system. The same framework also applies to min-max MPC.

An ISS framework is also applied to NTMPC in [113], reducing the problem into a nominal MPC optimisation with tightened constraints.

The use of deep learning to learn a robust NMPC policy has been proposed e.g. in [114] where a deep neural network is trained on pre-recorded data pairs generated from the offline solution of a multi-stage robust NMPC problem in which

the uncertainty is represented as a scenario tree. This approach, despite providing a very efficient online implementation, suffers from a lack of theoretical guarantees on convergence and stability and requires significant computations to be performed offline in order to train the deep neural network as the complexity of the optimisation problem explodes exponentially with the uncertainty discretisation. Deep learning is leveraged with a different mindset in [115] where tubes containing the trajectory of the system are learnt and then used in a NTMPC framework.

An approach based on contraction theory is proposed in [116] where the optimisation problem in the ancillary tracking MPC is augmented with a differential Lyapunov stability constraint on the control input. This condition guarantees exponential convergence of the closed-loop for any feasible nominal trajectory generated by the outer-loop MPC controller, which makes the proposed controller equilibrium-independent (i.e the controller does not need to be re-synthesised when the trajectory is updated). A feasibility recovery strategy is employed to make sure that the trajectory generated by the outer-loop MPC satisfies the Lyapunov constraint in order to guarantee the feasibility of the ancillary tracking problem. Moreover, feedback is incorporated in the outer-loop MPC algorithm via the contraction metric-weighted distance between the measured state and the initial nominal state treated as a decision variable in the optimisation problem. The two successive MPC controllers employ different sampling times and horizons: the outer-loop MPC scheme has a larger horizon and lower sampling rate by contrast to the ancillary controller with smaller horizon and higher sampling rate, allowing for faster disturbance rejection. The so-called differential Lyapunov-based TMPC is shown to achieve robust stability and fast response of the closed-loop.

Contraction theory is also leveraged in [117] to design a NTMPC algorithm for collision free trajectory generation and tracking of autonomous vehicles subject to bounded disturbances. A general feedback controller is designed offline using control contraction metrics (a differential-geometric generalisation of control Lyapunov functions) and the associated invariant tube is constructed with conservative ellipsoidal outer approximation of the uncertainty sets centred around any feasible

nominal trajectory. An optimal controller that minimises the tube cross-section is obtained by solving offline a quasiconvex program. Then, the tube is used by a motion planning algorithm as a robustness margin to tighten the constraint sets and generate online a collision free nominal trajectory; a local replanning procedure over a short-term horizon is also incorporated in an MPC fashion in order to reduce the tracking cost as new information on the disturbance realisation is available. Finally, robust trajectory tracking is implemented as usual with a two-tier control law consisting of the feedback controller generated offline via contraction theory and the open-loop nominal control sequence computed online by the MPC motion planner. The proposed robust controller offers the same important stability guarantees as for previous TMPC algorithms, while optimising a nonlinear feedback controller by contrast to approaches that treat nonlinearities as bounded disturbances, e.g. [83]. However, the method is conservative in the sense that the state dependent invariant sets associated with the control contraction metrics controller were approximated by ellipsoidal bounds to permit a tractable online implementation of the motion planner.

The approach is different in [118] where instead of computing a global controller and a tube that are valid for any feasible trajectory, a library of motion primitives (i.e elementary manoeuvres) is constructed offline and invoked online by a real-time planner. For each motion primitive of the library, a controller and a tube are precomputed leveraging tools from Lyapunov theory and sum-of-squares (SOS) programming. As usual, a two-tier control law is considered with a feedback controller and an open loop control sequence that generates the nominal trajectory. The latter is obtained by a direct collocation method. The feedback controller is parameterised by a polynomial, whose coefficients are optimised to minimise the sections of the tube (approximated by outer ellipsoidal sets). This is done by enforcing that a candidate Lyapunov function and its time derivative (that depends on the input polynomial coefficients) satisfy the usual positiveness and negativeness requirements, respectively. Such an optimisation problem, formulated as a SOS program, is solved for each motion primitive, thus populating a library of controllers and tubes associated with each elementary manoeuvre. Then, a real-time scheduler

browses the library to sequentially combine compatible motion primitives and generate a collision free trajectory. The methodology is demonstrated for various dynamical systems (ground vehicle, quadrotor and fixed-wing aircraft) and relatively aggressive manoeuvres could be achieved in the lab while ensuring collision-free trajectories. The drawback of the proposed approach is that the offline optimisation problem is bilinear and an alternation scheme has to be used where the problem is solved in several steps by alternatively forcing a subset of the decision variables to constant values, with no guarantee that the global optimum is reached. Bilinearity of the problem in the controller parameters and Lyapunov function prevent these optimisation variables to be searched at the same time, which does not truly guarantee that the computed controller achieves a minimum-size tube. Although an additional alternation step is added to improve the quality of the designed controllers and tubes, no guarantee of convergence towards the global optimum is given. Another drawback arises from the somewhat conservative choice of ellipsoidal sets for the tube parameterisation. Finally, the method requires significant offline computations for populating the library and there is a trade-off between increasing the size of the library to offer richer planning capabilities and the time needed by the real-time planner to select one motion primitive in a larger catalogue.

Most of the frameworks described above considered a parameterisation of the control law in the form of a two-degree-of-freedom controller. The approach is different in [119] where a min-max differential inequality is derived to construct a robust forward invariant tube and the associated NTMPC strategy for nonlinear systems with input-affine structure. The method is tractable since it scales linearly with the horizon length.

2.4.4 Identified gap

Although TMPC is fairly well understood for linear systems, many challenges need to be addressed in the context of nonlinear systems. The main one being that nonlinear TMPC is still numerically intractable for general nonlinear systems. Approaches based on successive linearisation and treating the linearisation error as a

bounded disturbance are promising, but they do not solve the problem of bounding the error conservatively. This could be addressed by convexification of the system dynamics (by means of difference of convex (DC) function decompositions), allowing to bound the linearisation error tightly. Moreover, application of these techniques to robust control of tiltwing VTOL aircraft would represent a novelty and would enable reliable and efficient operation of these vehicles in safety-critical scenarios.

3

Predictive energy management for hybrid-electric aircraft

Contents

3.1 Introduction

Electric propulsion systems are a vital part of the technology roadmap for decarbonising the air transport industry. The commercial viability of aircraft with electric or hybrid-electric propulsion and the effectiveness of this approach for reducing emissions relies on onboard energy management. This is due to the fundamental limits that are imposed by low energy density of electrical energy storage devices in current and likely future battery technology. Control and optimisation strategies face numerous challenges in this context because of the complexity of powertrain models, which typically contain empirical and physics-based subsystems, as well as the demanding nature of optimising predicted trajectories and energy usage. Additional challenges are the need for reliable, accurate and efficient solution methods. In this chapter, we will address these issues by developing an energy management strategy leveraging convex optimisation for a new generation of more electric aircraft. The presented algorithm meets the requirements in terms of optimality and computational tractability.

We present a model predictive control (MPC) algorithm for energy management in aircraft with hybrid electric propulsion systems consisting of gas turbine and electric motor components. Series and parallel configurations are considered. By combining a point-mass aircraft dynamical model with models of electrical losses and losses in the gas turbine, the fuel consumed over a given future flight path is minimised subject to constraints on the battery, electric motor and gas turbine. The optimisation is formulated as a convex problem under mild assumptions and its solution is used to define a predictive energy management control law that takes into account the variation in aircraft mass during flight. We investigate the performance of algorithms for solving this problem. An alternating direction method of multipliers (ADMM) algorithm is proposed and compared with a general purpose convex interior point solver. We also show that the ADMM implementation reduces the required computation time by orders of magnitude in comparison with

a general purpose convex programming solver, making it suitable for real-time supervisory energy management control.

The chapter is organised as follows. Mathematical models of the powertrain components and aircraft dynamics are developed in Section 3.2. Energy management problems for series and parallel configurations are stated as receding-horizon optimisation problems in Section 3.3. Section 3.4 presents a series of simplifications that yield convex relaxations of these problems. In particular, a unified formulation is proposed for both powertrain configurations. The ADMM solver is presented in Section 3.5 and its performance and potential for real-time implementation are discussed in Section 3.6. Conclusions are presented in Section 3.7.

3.2 Modeling

This section derives models of the aircraft dynamics and powertrain components (battery, electric motor, gas turbine etc.), which will be used to formulate the energy management problem as a model-based optimisation problem.

We consider a hybrid electric aircraft propulsion system with either a series or parallel topology. When the power output demand is negative, which may occur for example while the aircraft is descending, we consider the possibility of using the same powertrain to generate electrical energy (i.e. operating in a "windmilling" mode) in order to recharge the battery. In practice a variable-pitch fan would be required for this functionality, which would increase complexity.

3.2.1 Aircraft dynamics

The aircraft motion is constrained by its dynamic equations. Assuming a point-mass model [120] and referring to Figure 3.1, the equilibrium of forces yields

$$m\frac{\mathrm{d}}{\mathrm{d}t}(\overrightarrow{v}) = \overrightarrow{T} + \overrightarrow{L} + \overrightarrow{D} + \overrightarrow{W},$$

where $\overrightarrow{v}$ is the velocity vector, m the instantaneous mass of the aircraft, $\overrightarrow{T}$ the vector of thrust, $\overrightarrow{L}$ and $\overrightarrow{D}$ are the lift and drag vectors and $\overrightarrow{W}$ is the aircraft weight.

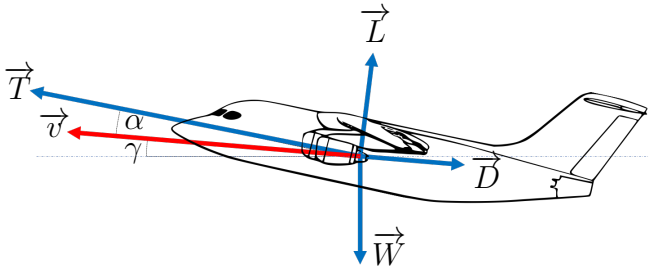


Figure 3.1: Aircraft forces and motion.

Using the coordinates (v,γ), where v is the velocity vector magnitude and γ is the flight path angle, and projecting the vector equation in wind axes along the drag vector $\overrightarrow{D}$ yields

$$m\frac{\mathrm{d}}{\mathrm{d}t}v + mg\sin\gamma = T\cos\alpha - \frac{1}{2}C_D\rho S v^2.$$

Here S is the wing area, ρ is the density of air, g is acceleration due to gravity, $C_D = C_D(\alpha)$ the drag coefficient and α the angle of attack. Projecting along the lift vector $\overrightarrow{L}$ yields

$$mv\frac{\mathrm{d}}{\mathrm{d}t}\gamma + mg\cos\gamma = T\sin\alpha + \frac{1}{2}C_L\rho S v^2,$$

where $C_L = C_L(\alpha)$ is the lift coefficient.

The drive power is given as follows

$$P_{\mathrm{drv}} = \overrightarrow{T}\cdot\overrightarrow{v} = m\frac{\mathrm{d}}{\mathrm{d}t}(\frac{1}{2}v^2) + \frac{1}{2}C_D\rho S v^3 + mgv\sin\gamma.$$

3.2.2 Hybrid propulsion system

Parallel architecture

In the parallel architecture ($\mathcal{T} = \mathcal{P}$), a gas turbine producing power P_{gt} is mechanically coupled with an electric motor with power output P_{em} in a parallel arrangement (Fig. 3.2). These two power sources are combined to give the power output of the propulsion system, P_{drv}, via

$$P_{\mathrm{drv}}(t) = P_{\mathrm{gt}}(t) + P_{\mathrm{em}}(t),$$

where 100% efficiency in drivetrain components is assumed.

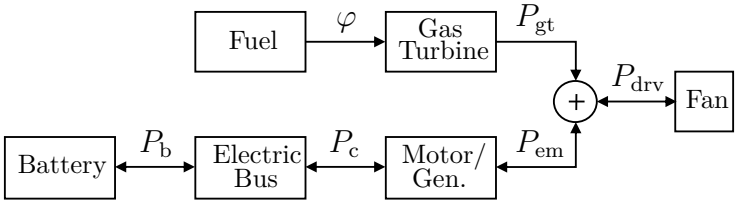


Figure 3.2: Parallel-hybrid propulsion architecture.

Series architecture

In the series architecture ($\mathcal{T} = \mathcal{S}$), the propulsion system power output P_{drv} is delivered by an electric motor taking electrical power P_{el} from two sources: a battery with effective power output P_{c} and a turbo generator set (gas turbine in series with an electric generator) with power output P_{gen} (Fig. 3.3). The power balance is given by

$$P_{\text{el}}(t) = P_{\text{gen}}(t) + P_{\text{c}}(t).$$

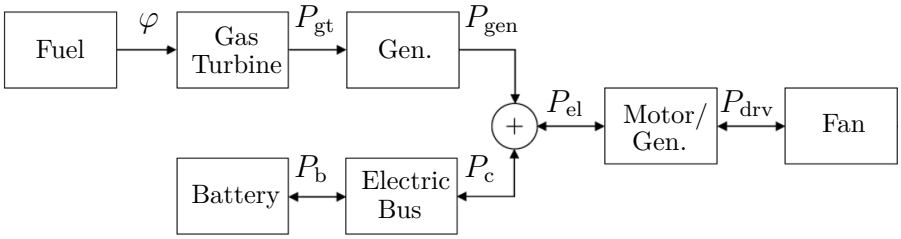


Figure 3.3: Series-hybrid propulsion architecture.

3.2.3 Battery

The battery is modeled as an equivalent circuit with internal resistance R and open-circuit voltage U, so that the input-output map between its chemical power P_{b} and the effectively delivered electrical power P_{c} is given by [121]

$$\begin{aligned} P_{\text{b}} &= g\big(P_{\text{c}}\big), \\ &= \frac{U^2}{2R}\left(1 - \sqrt{1 - \frac{4R}{U^2}P_{\text{c}}}\right), \end{aligned}$$

where U and R are assumed constant [51]. The evolution of the battery state of charge (SOC) $E(t)$ is given by

$$\dot{E} = -P_{\text{b}} \tag{3.1}$$

and $E(t)$ is subject at all times to upper and lower bounds

$$\underline{E} \leq E \leq \overline{E}.$$

3.2.4 Gas turbine

The rate of change of mass of the aircraft is given by

$$\dot{m} = -\varphi = -f(P_{\mathrm{gt}}(t), \omega_{\mathrm{gt}}(t)), \tag{3.2}$$

where φ is the rate of fuel consumption and $f(P_{\mathrm{gt}}, \omega_{\mathrm{gt}})$ is a piecewise-quadratic function of the gas turbine power output P_{gt} and shaft rotation speed ω_{gt}. We assume that $f(\cdot)$ can be determined empirically from fuel map data in the form

$$\begin{aligned}\varphi &= f(P_{\mathrm{gt}}, \omega_{\mathrm{gt}}),\\ &= \beta_2(\omega_{\mathrm{gt}})P_{\mathrm{gt}}^2 + \beta_1(\omega_{\mathrm{gt}})P_{\mathrm{gt}} + \beta_0(\omega_{\mathrm{gt}}),\end{aligned}$$

with $\beta_2(\omega_{\mathrm{gt}}) \geq 0$ and $\beta_1(\omega_{\mathrm{gt}}) > 0$ in the operating range of ω_{gt}. The power P_{gt} and shaft rotation speed ω_{gt} are limited by

$$\begin{aligned}\underline{P}_{\mathrm{gt}} \leq P_{\mathrm{gt}} \leq \overline{P}_{\mathrm{gt}},\\ \underline{\omega}_{\mathrm{gt}} \leq \omega_{\mathrm{gt}} \leq \overline{\omega}_{\mathrm{gt}}.\end{aligned}$$

These limits apply to both parallel and series configurations, and in the latter case they constrain the turbo generator set.

3.2.5 Electric motor

In the parallel configuration, the electric motor input-output map between input electrical power P_{c} and effective mechanical power P_{em} is modeled by a piecewise-quadratic function

$$\begin{aligned}P_{\mathrm{c}} &= h(P_{\mathrm{em}}(t), \omega_{\mathrm{em}}(t)),\\ &= \kappa_2(\omega_{\mathrm{em}})P_{\mathrm{em}}^2 + \kappa_1(\omega_{\mathrm{em}})P_{\mathrm{em}} + \kappa_0(\omega_{\mathrm{em}}),\end{aligned}$$

where ω_{em} is the electric motor shaft rotation speed and $\kappa_2(\omega_{\mathrm{em}}) \geq 0$, $\kappa_1(\omega_{\mathrm{em}}) > 0$ for all ω_{em} in the operating range. The function $h(\cdot)$ can be determined empirically

from electric motor loss data. The limitations on the electric motor power and shaft rotation speeds are set by the following constraints

$$\underline{P}_{\text{em}} \leq P_{\text{em}} \leq \overline{P}_{\text{em}},$$
$$\underline{\omega}_{\text{em}} \leq \omega_{\text{em}} \leq \overline{\omega}_{\text{em}}.$$

In the series configuration, the input-output map between the input electrical power P_{el} and effective mechanical power P_{drv} is modeled by $P_{\text{el}} = h(P_{\text{drv}}(t), \omega_{\text{drv}}(t))$, where ω_{drv} is the fan shaft rotation speed. The limitations on the electric motor power and shaft rotation speeds are set by the following constraints

$$\underline{P}_{\text{drv}} \leq P_{\text{drv}} \leq \overline{P}_{\text{drv}},$$
$$\underline{\omega}_{\text{drv}} \leq \omega_{\text{drv}} \leq \overline{\omega}_{\text{drv}}.$$

3.2.6 Electric generator

In the series configuration a generator converts the gas turbine mechanical power P_{gt} into electrical power P_{gen}. This electrical machine is modeled by a piecewise-quadratic function

$$\begin{aligned} P_{\text{gt}} &= h_{\text{gen}}(P_{\text{gen}}(t), \omega_{\text{gen}}(t)), \\ &= \nu_2(\omega_{\text{gen}}) P_{\text{gen}}^2 + \nu_1(\omega_{\text{gen}}) P_{\text{gen}} + \nu_0(\omega_{\text{gen}}), \end{aligned}$$

where ω_{gen} is the electric generator shaft rotation speed and $\nu_2(\omega_{\text{gen}}) \geq 0$, $\nu_1(\omega_{\text{gen}}) > 0$ for all ω_{gen} in the operating range. The loss map $h_{\text{gen}}(\cdot)$ can be determined empirically from electric generator loss data. The limits on power and shaft rotation speed for the electric generator are encapsulated by the inequality constraints given for the gas turbine.

3.2.7 Objective

The problem at hand is to find the real-time optimal power split between the gas turbine and electric motor that minimises

$$J = \int_0^T f(P_{\text{gt}}(t), \omega_{\text{gt}}(t)) \mathrm{d}t,$$

while satisfying constraints on the battery SOC and limits on power flows throughout the powertrain, and while producing sufficient power to follow a prescribed flight path.

3.3 Discrete-time optimal control

This section describes a discrete-time model that enables the optimal power split between battery and fuel over a given future flight path to be determined as a finite-dimensional optimisation problem. For a fixed sampling interval δ, we consider a predictive control strategy that minimises, at each sampling instant, the fuel consumption over the remaining flight path. The optimisation is performed subject to the dynamics of the aircraft mass and the battery SOC. The problem is also subject to limits on energy stored in the battery (to prevent deep discharging or overcharging) and limits on power flows corresponding to physical and safety constraints.

The optimal solution to the fuel minimisation problem at the kth sampling instant is computed using estimates of the battery SOC $E(k\delta)$ and the aircraft mass $m(k\delta)$, so that $E_0 = E(k\delta)$ and $m_0 = m(k\delta)$ at any time $k\delta$. The control law at time $k\delta$ is defined by the first time step of this optimal solution. The notation $\{x_0, x_1, \dots x_{N-1}\}$ is used for the sequence of current and future values of a variable x predicted at the kth discrete-time step, so that x_i denotes the predicted value of $x\Big((k+i)\delta\Big)$. The horizon N is chosen so that $N = \lceil T/\delta \rceil - k$, and hence N shrinks as k increases and $k\delta$ approaches T.

The discrete-time approximation of the objective is

$$J = \sum_{i=0}^{N-1} f_i(P_{\mathrm{gt},i}, \omega_{\mathrm{gt},i})\, \delta, \tag{3.3}$$

with, for $i = 0, \dots, N-1$,

$$f_i(P_{\mathrm{gt},i}, \omega_{\mathrm{gt},i}) = \beta_2(\omega_{\mathrm{gt},i}) P_{\mathrm{gt},i}^2 + \beta_1(\omega_{\mathrm{gt},i}) P_{\mathrm{gt},i} + \beta_0(\omega_{\mathrm{gt},i}), \tag{3.4}$$

$$m_{i+1} = m_i - f_i(P_{\mathrm{gt},i}, \omega_{\mathrm{gt},i})\, \delta, \tag{3.5}$$

where the forward Euler approximation has been used to discretise (3.2). The same approach applied to (3.1) yields the discrete-time battery model

$$E_{i+1} = E_i - P_{\mathrm{b},i}\,\delta, \tag{3.6}$$

$$\begin{aligned} P_{\mathrm{b},i} &= g_i(P_{\mathrm{c},i}), \\ &= \frac{U^2}{2R}\left[1 - \sqrt{1 - \frac{4R}{U^2}P_{\mathrm{c},i}}\right], \end{aligned} \tag{3.7}$$

for $i = 0, \ldots, N-1$. In the parallel configuration, the electric motor input-output map is given by

$$\begin{aligned} P_{\mathrm{c},i} &= h_i(P_{\mathrm{em},i}, \omega_{\mathrm{em},i}), \\ &= \kappa_2(\omega_{\mathrm{em},i})P_{\mathrm{em},i}^2 + \kappa_1(\omega_{\mathrm{em},i})P_{\mathrm{em},i} + \kappa_0(\omega_{\mathrm{em},i}), \end{aligned} \tag{3.8}$$

while for the series configuration we have

$$\begin{aligned} P_{\mathrm{el},i} &= h_i(P_{\mathrm{drv},i}, \omega_{\mathrm{drv},i}), \\ &= \kappa_2(\omega_{\mathrm{drv},i})P_{\mathrm{drv},i}^2 + \kappa_1(\omega_{\mathrm{drv},i})P_{\mathrm{drv},i} + \kappa_0(\omega_{\mathrm{drv},i}), \end{aligned} \tag{3.9}$$

and

$$\begin{aligned} P_{\mathrm{gt},i} &= h_{\mathrm{gen},i}(P_{\mathrm{gen},i}, \omega_{\mathrm{gen},i}), \\ &= \nu_2(\omega_{\mathrm{gen},i})P_{\mathrm{gen},i}^2 + \nu_1(\omega_{\mathrm{gen},i})P_{\mathrm{gen},i} + \nu_0(\omega_{\mathrm{gen},i}). \end{aligned} \tag{3.10}$$

The aircraft dynamics are given in discrete time by

$$m_i v_i \Delta_i\gamma + m_i g\cos(\gamma_i) = T_i\sin(\alpha_i) + \tfrac{1}{2}C_L(\alpha_i)\rho S v_i^2 \tag{3.11}$$

$$P_{\mathrm{drv},i} = \tfrac{1}{2}m_i\Delta_i(v^2) + m_i g\sin(\gamma_i)v_i + \tfrac{1}{2}C_D(\alpha_i)\rho S v_i^3 \tag{3.12}$$

for $i = 0, \ldots, N-1$, where

$$\Delta_i(v^2) = (v_{i+1}^2 - v_i^2)/\delta, \quad \Delta_i\gamma = (\gamma_{i+1} - \gamma_i)/\delta.$$

The power balance in discrete time for the parallel and series case respectively is given by

$$P_{\mathrm{drv},i} = P_{\mathrm{gt},i} + P_{\mathrm{em},i}, \tag{3.13}$$

$$P_{\mathrm{el},i} = P_{\mathrm{c},i} + P_{\mathrm{gen},i}. \tag{3.14}$$

3.3.1 Parallel architecture

For the parallel architecture the problem solved at the kth time step is

$$\min_{\substack{P_{\text{gt}},P_{\text{em}},P_{\text{drv}},m,\\ E,\omega_{\text{gt}},\omega_{\text{em}},\alpha}} \quad \sum_{i=0}^{N-1} f_i(P_{\text{gt},i},\omega_{\text{gt},i})\delta \tag{3.15}$$

$$\begin{aligned}
\text{s.t.},\forall i \in [0,\dots,N-1], \quad & P_{\text{drv},i} = P_{\text{gt},i} + P_{\text{em},i} \\
& P_{\text{drv},i} = \tfrac{1}{2} m_i \Delta_i v^2 + m_i g \sin(\gamma_i) v_i + \tfrac{1}{2} C_D(\alpha_i) \rho S v_i^3 \\
& m_i v_i \Delta_i \gamma + m_i g \cos\gamma_i = T_i \sin\alpha_i + \tfrac{1}{2} C_L(\alpha_i) \rho S v_i^2 \\
& m_{i+1} = m_i - f_i(P_{\text{gt},i},\omega_{\text{gt},i})\,\delta \\
& E_{i+1} = E_i - g_i\left(h_i\left(P_{\text{em},i},\omega_{\text{em},i}\right)\right)\,\delta \\
& m_0 = m(k\delta) \\
& E_0 = E(k\delta) \\
& \underline{E} \le E_i \le \overline{E} \\
& \underline{P}_{\text{gt}} \le P_{\text{gt},i} \le \overline{P}_{\text{gt}} \\
& \underline{\omega}_{\text{gt}} \le \omega_{\text{gt},i} \le \overline{\omega}_{\text{gt}} \\
& \underline{P}_{\text{em}} \le P_{\text{em},i} \le \overline{P}_{\text{em}} \\
& \underline{\omega}_{\text{em}} \le \omega_{\text{em},i} \le \overline{\omega}_{\text{em}}
\end{aligned}$$

3.3.2 Series architecture

For the series architecture, the problem solved at the kth time step is

$$\min_{\substack{P_{\text{gt}},P_{\text{el}},P_{\text{drv}},P_{\text{gen}},P_{\text{c}},\\ m,E,\omega_{\text{gt}},\omega_{\text{drv}},\alpha}} \quad \sum_{i=0}^{N-1} f_i(P_{\text{gt},i},\omega_{\text{gt},i})\delta \tag{3.16}$$

$$\begin{aligned}
\text{s.t.},\forall i \in [0,\dots,N-1], \quad & P_{\text{el},i} = P_{\text{c},i} + P_{\text{gen},i} \\
& P_{\text{drv},i} = \tfrac{1}{2} m_i \Delta_i v^2 + m_i g \sin(\gamma_i) v_i + \tfrac{1}{2} C_D(\alpha_i) \rho S v_i^3 \\
& m_i v_i \Delta_i \gamma + m_i g \cos\gamma_i = T_i \sin\alpha_i + \tfrac{1}{2} C_L(\alpha_i) \rho S v_i^2 \\
& m_{i+1} = m_i - f_i(P_{\text{gt},i},\omega_{\text{gt},i})\,\delta \\
& E_{i+1} = E_i - g_i(P_{\text{c},i})\,\delta \\
& P_{\text{el},i} = h_i(P_{\text{drv},i},\omega_{\text{drv},i})
\end{aligned}$$

$$P_{\mathrm{gt},i} = h_{\mathrm{gen},i}(P_{\mathrm{gen},i}, \omega_{\mathrm{gen},i})$$

$$m_0 = m(k\delta)$$

$$E_0 = E(k\delta)$$

$$\underline{E} \leq E_i \leq \overline{E}$$

$$\underline{P}_{\mathrm{gt}} \leq P_{\mathrm{gt},i} \leq \overline{P}_{\mathrm{gt}}$$

$$\underline{\omega}_{\mathrm{gt}} \leq \omega_{\mathrm{gt},i} \leq \overline{\omega}_{\mathrm{gt}}$$

$$\underline{P}_{\mathrm{drv}} \leq P_{\mathrm{drv},i} \leq \overline{P}_{\mathrm{drv}}$$

$$\underline{\omega}_{\mathrm{drv}} \leq \omega_{\mathrm{drv},i} \leq \overline{\omega}_{\mathrm{drv}}$$

3.4 Convex relaxation

The optimisation problems in (3.15) and (3.16) are nonconvex, which makes a real-time implementation of an MPC algorithm that relies on its solution computationally intractable. In this section a convex approximation is proposed that is suitable for an online solution. We make three simplifications: 1) we prescribe a flight profile and impose an assumption on the monotonicity of the loss map functions which results in convex loss map functions and allows their coefficients to be computed a priori; 2) we reformulate the dynamics as a quadratic function of aircraft mass under mild assumptions; 3) we introduce a lossless change of optimisation variables that shifts the nonlinear term in the battery update equation to the power balance inequality.

3.4.1 Reformulation of the loss map functions

We assume that the aircraft speed v_i and flight path angle γ_i are chosen externally by a suitable guidance algorithm for $i = 0, \ldots, N-1$. This assumption is reasonable for an actual air traffic management application where flight corridors are prescribed. For the series configuration, we assume that the generator speed is constant: $\omega_{\mathrm{gen},i} = \omega^*_{\mathrm{gen}}$, $\forall i$, where the optimal speed ω^*_{gen} is determined empirically so as to operate the turbo generator set at its maximum efficiency. This allows us to

fix the coefficients in (3.10) to constant values and express $h_{\text{gen},i}(P_{\text{gen},i}, \omega_{\text{gen},i})$ as a convex quadratic function of $P_{\text{gen},i}$

$$h_{\text{gen},i}(P_{\text{gen},i}) = \nu_2 P_{\text{gen},i}^2 + \nu_1 P_{\text{gen},i} + \nu_0, \tag{3.17}$$

with constant coefficients $\nu_2 \geq 0,\ \nu_1 > 0$.

For the parallel configuration, we assume for simplicity that the gas turbine, electric motor and fan share a common shaft rotation speed, i.e. $\omega_{\text{gt},i} = \omega_{\text{em},i} = \omega_{\text{drv},i}$, $\forall i$. If the fan shaft speed is known at each time step of the prediction horizon, then the coefficients in (3.8) and (3.9) can be estimated from a set of polynomial approximations of $h_i(\cdot)$ at a pre-determined set of speeds. This allows $h_i(P_{\text{em},i}, \omega_{\text{em},i})$ and $h_i(P_{\text{drv},i}, \omega_{\text{drv},i})$ to be replaced by time-varying convex functions of power alone

$$h_i(P_{\text{em},i}) = \kappa_{2,i} P_{\text{em},i}^2 + \kappa_{1,i} P_{\text{em},i} + \kappa_{0,i}, \tag{3.18}$$

$$h_i(P_{\text{drv},i}) = \kappa_{2,i} P_{\text{drv},i}^2 + \kappa_{1,i} P_{\text{drv},i} + \kappa_{0,i}, \tag{3.19}$$

with $\kappa_{2,i} \geq 0$, $\kappa_{1,i} > 0$, $\kappa_{2,i} \geq 0$, $\kappa_{1,i} > 0$, for all i. Regarding the gas turbine fuel map, since the spool speed is assumed constant, the coefficients are independent of gas turbine spool speed such that $f_i(P_{\text{gt},i}, \omega_{\text{gt},i})$ can also be replaced by a convex functions of power alone

$$f_i(P_{\text{gt},i}) = \beta_2 P_{\text{gt},i}^2 + \beta_1 P_{\text{gt},i} + \beta_0, \tag{3.20}$$

with $\beta_2 \geq 0,\ \beta_1 > 0$.

Since these functions are loss maps, we formulate the following assumption, as suggested in [51].

Assumption 1 (Non-decreasingness of the loss maps)**.** *The functions $h_{gen,i}$, f_i, h_i are non-decreasing.*

It follows from Assumption 1 that: $P_{\text{em},i} \geq -\kappa_{1,i}/2\kappa_{2,i}$, $P_{\text{drv},i} \geq -\kappa_{1,i}/2\kappa_{2,i}$, $P_{\text{gen},i} \geq -\nu_1/2\nu_2$, $P_{\text{gt},i} \geq -\beta_1/2\beta_2$ for all i, which requires new lower bounds on power. In the parallel configuration, the new bounds are given by

$$\underline{P}_{\text{em},i} = \max\left\{\underline{P}_{\text{em}}, -\frac{\kappa_{1,i}}{2\kappa_{2,i}}\right\}, \tag{3.21}$$

$$\underline{P}_{\mathrm{gt}} = \max\left\{\underline{P}_{\mathrm{gt}}, -\frac{\beta_1}{2\beta_2}\right\}, \tag{3.22}$$

whereas in the series configuration only the gas turbine bound should be updated, as follows

$$\underline{P}_{\mathrm{gt}} = \max\left\{\underline{P}_{\mathrm{gt}}, -\frac{\beta_1}{2\beta_2}, h_{\mathrm{gen},i}\left(-\frac{\nu_1}{2\nu_2}\right)\right\}, \tag{3.23}$$

since we can enforce the monotonicity condition on the drive power a priori when prescribing the drive power profile.

In order to estimate the shaft speed $\omega_{\mathrm{drv},i}$, and hence determine the coefficients in (3.18)-(3.20), we use a pre-computed look-up table relating the drive power to rotational speed of the fan, for a given altitude, Mach number, and air conditions (temperature and specific heat at constant pressure). This enables the shaft speed to be determined as a function of the fan power output at each discrete-time step along the flight path. Although $P_{\mathrm{drv},i}$ depends on the aircraft mass m_i, which is itself an optimisation variable, a prior estimate of the required power output can be obtained by assuming a constant mass $m_i = m_0$ for all i.

Note that since the rotation speeds are prescribed, all constraints on shaft rotation speeds can be removed from the optimisation (and checked a priori). The same remark holds for the constraints on the drive power.

3.4.2 Reformulation of the dynamics

To express the dynamics in a form suitable for convex programming, we simplify the dynamical equations and combine the equations that constrain the aircraft motion as follows. First we express the drag and lift coefficients, C_D and C_L, as functions of the angle of attack α. Over a restricted domain and for given Reynolds and Mach numbers, the drag and lift coefficients can be expressed respectively as a quadratic non-decreasing function and a linear non-decreasing function [122]

$$C_D(\alpha_i) = a_2\alpha_i^2 + a_1\alpha_i + a_0, \qquad a_2 > 0, \tag{3.24}$$

$$C_L(\alpha_i) = b_1\alpha_i + b_0, \qquad b_1 > 0, \tag{3.25}$$

for $\underline{\alpha} \leq \alpha_i \leq \overline{\alpha}$. Secondly, assuming that the contribution of the thrust in the vertical direction is negligible[1], the term $T\sin(\alpha)$ can be neglected from (3.11). Finally, combining (3.11), (3.12), (3.24) and (3.25), the angle of attack can be eliminated from the expression for $P_{\mathrm{drv},i}$, which can be expressed as a quadratic function of the aircraft mass, m_i, as follows

$$P_{\mathrm{drv},i} = \eta_{2,i} m_i^2 + \eta_{1,i} m_i + \eta_{0,i}, \tag{3.26}$$

where

$$\begin{aligned}
\eta_{2,i} &= \frac{2a_2(v_i\Delta_i\gamma + g\cos\gamma_i)^2}{b_1^2\rho S v_i},\\
\eta_{1,i} &= \tfrac{1}{2}\Delta_i v^2 + g\sin\gamma_i v_i - \frac{2a_2 b_0(v_i\Delta_i\gamma + g\cos\gamma_i)v_i}{b_1^2} + \frac{a_1}{b_1}(v_i\Delta_i\gamma + g\cos\gamma_i)v_i,\\
\eta_{0,i} &= \tfrac{1}{2}\rho S v_i^3\Big(\frac{a_2 b_0^2}{b_1^2} - \frac{a_1 b_0}{b_1} + a_0\Big).
\end{aligned}$$

Since the flight path angle γ_i and speed v_i are determined a priori, the coefficients $\eta_{0,i}$, $\eta_{1,i}$, $\eta_{2,i}$ are fixed. Moreover $\eta_{2,i} > 0$ for all i, so the drive power is a convex function of m_i. Note that there is no guarantee that satisfaction of (3.26) enforces (3.11) and (3.12) individually. In practice, assuming that we have full control over the eliminated variable α (via the elevator and fans), both individual dynamical equations can be satisfied a posteriori. The existence of a faster inner flight control loop on angle of attack and thrust ensures that the correct trajectory is followed. The bounds $\underline{\alpha} \leq \alpha_i \leq \overline{\alpha}$ need to be checked a posteriori.

3.4.3 Reformulation of power balance

Let the rate of change of fuel mass be $\varphi_i := f_i(P_{\mathrm{gt},i})$. Using this new variable, the power balance can be enforced by

$$\varphi_i = f_{\varphi,i}(m_i, P_{\mathrm{b},i}),$$

where the function $f_{\varphi,i}$ is defined

$$f_{\varphi,i} = \begin{cases} f_i\left(P_{\mathrm{drv},i}(m_i) - h_i^{-1}\left(g_i^{-1}\left(P_{\mathrm{b},i}\right)\right)\right) & \text{if } \mathcal{T} = \mathcal{P} \\ f_i\left(h_{\mathrm{gen},i}\left(h_i\left(P_{\mathrm{drv},i}(m_i)\right) - g_i^{-1}\left(P_{\mathrm{b},i}\right)\right)\right) & \text{if } \mathcal{T} = \mathcal{S} \end{cases} \tag{3.27}$$

[1]This assumption was checked in simulations, where it was found that the solution satisfies $\alpha < 2^\circ$, which supports this assumption.

where $P_{\mathrm{drv},i}$ is given by (3.26). This formulation unifies the treatment of series and parallel configurations and eliminates the variables P_{el}, P_{gen} and P_{em} from the optimisation problem. Moreover, by the following lemma and theorem, the function $f_{\varphi,i}(\cdot)$ is convex.

Lemma 1 (Concave inverse map)**.** *Let g be a convex, twice differentiable, non-decreasing, one-to-one function. Then the function g^{-1} is concave increasing.*

Proof. Since g is one-to-one, if $y = g(x)$ then $x = g^{-1}(y)$ is unique $\forall x$. Differentiating twice yields $dx/dy = 1/(g'(x))$ and $d^2x/dy^2 = -g''/(g'(x))^2 dx/dy = -g''/(g'(x))^3$. Since g is non-decreasing $g' \geq 0$ and thus $dx/dy = 1/(g'(x)) > 0$ which implies that g^{-1} is monotonically increasing. Moreover, since g is convex, $g'' \geq 0$ and $d^2x/dy^2 = -g''/(g'(x))^3 \leq 0$. By the second order convexity condition 11, Appendix A, it follows that g^{-1} is concave. □

Theorem 2 (Convexity of $f_{\varphi,i}(\cdot)$)**.** *The following functions are convex:*

(a) $\varphi_i(m_i, P_{b,i}) = f_i\left(P_{drv,i}(m_i) - h_i^{-1}\left(g_i^{-1}\left(P_{b,i}\right)\right)\right)$,

(b) $\varphi_i(m_i, P_{b,i}) = f_i\left(h_{gen,i}\left(h_i\left(P_{drv,i}(m_i)\right) - g_i^{-1}\left(P_{b,i}\right)\right)\right)$.

Proof. Note first that from equations (3.7), (3.17)-(3.20) and (3.26), the functions $g_i(\cdot)$, $h_{\mathrm{gen},i}(\cdot)$, $h_i(\cdot)$, $f_i(\cdot)$, $P_{\mathrm{drv},i}(\cdot)$ are convex, twice differentiable, one-to-one functions. Moreover, from Assumption 1, the functions $h_{\mathrm{gen},i}(\cdot)$, $h_i(\cdot)$, $f_i(\cdot)$ are also non-decreasing. Consider both cases separately:

- (a) From lemma 1 and by the properties of g and h, the inverse maps g_i^{-1}, h_i^{-1} are concave increasing (and in particular non-decreasing). Thus, the composition $h_i^{-1} \circ g_i^{-1}$ is concave (see Remark 11, Appendix A). By convexity of $P_{\mathrm{drv},i}$, the intermediate function $\phi_i = P_{\mathrm{drv},i}(m_i) - h_i^{-1}\left(g_i^{-1}\left(P_{\mathrm{b},i}\right)\right)$ is convex. The composition $f_i \circ \phi_i$ is convex since f_i is convex non-decreasing.

- (b) Since h_i is convex non-decreasing and $P_{\mathrm{drv},i}$ is convex, the composition $h_i \circ P_{\mathrm{drv},i}$ is convex (see Remark 11, Appendix A). From lemma 1 and by the properties of g, the inverse map g_i^{-1} is concave and $-g_i^{-1}$ is convex. Thus, the

intermediate function $\phi_i = h_i\left(P_{\mathrm{drv},i}(m_i)\right) - g_i^{-1}\left(P_{\mathrm{b},i}\right)$ is convex. Convexity of φ_i follows from the successive compositions of convex functions $f_i \circ h_{\mathrm{gen},i} \circ \phi_i$ where $f_i, h_{\mathrm{gen},i}$ are non-decreasing convex.

□

We can construct a convex program by relaxing the power balance equality to the inequality

$$\varphi_i \geq f_{\varphi,i}(m_i, P_{\mathrm{b},i}), \tag{3.28}$$

which is necessarily satisfied with equality at the optimum since the form of the objective in (3.15) and (3.16) ensures that any feasible solution that does not satisfy this constraint with equality is suboptimal.

The constraints on gas turbine power and electric motor power are replaced by constraints on rate of change of fuel mass and on battery power, respectively,

$$\underline{\varphi}_i \leq \varphi_i \leq \overline{\varphi}_i, \tag{3.29}$$

$$\underline{P}_{\mathrm{b},i} \leq P_{\mathrm{b},i} \leq \overline{P}_{\mathrm{b},i}, \tag{3.30}$$

with $\underline{\varphi}_i = f_i(\underline{P}_{\mathrm{gt}})$, $\overline{\varphi}_i = f_i(\overline{P}_{\mathrm{gt}})$. Here

$$\underline{P}_{\mathrm{b},i} = \begin{cases} g_i(h_i(\underline{P}_{\mathrm{em},i})) & \text{if } \mathcal{T} = \mathcal{P} \\ \widetilde{f}_i(\underline{P}_{\mathrm{em}}, \overline{P}_{\mathrm{gt}}) & \text{if } \mathcal{T} = \mathcal{S} \end{cases}$$

where $\widetilde{f}_i(x, y) := g_i(h_i(x) - h_{\mathrm{gen},i}^{-1}(y))$. Furthermore, to ensure that $g_i(\cdot)$ is real-valued we require $P_{\mathrm{c},i} \leq U^2/4R$, and hence

$$\overline{P}_{\mathrm{b},i} = \begin{cases} g_i(h_i(\overline{P}_{\mathrm{em},i})) & \text{if } \mathcal{T} = \mathcal{P} \\ \min\left\{\widetilde{f}_i(\overline{P}_{\mathrm{em}}, \underline{P}_{\mathrm{gt}}), \frac{U^2}{2R}\right\} & \text{if } \mathcal{T} = \mathcal{S} \end{cases}$$

where $\overline{P}_{\mathrm{em},i}$ is defined for the parallel configuration by

$$\overline{P}_{\mathrm{em},i} = \min\left\{\overline{P}_{\mathrm{em}}, r_{\max,i}\right\}$$

with $r_{\max,i} = \max\left\{x : 1 - 4R/U^2 h_i(x) = 0\right\}$.

3.4.4 Convex program

A unified convex program can thus be formulated as follows

$$\begin{aligned}
\min_{\varphi,\, P_\mathrm{b},\, m,\, E} \quad & \sum_{i=0}^{N-1} \varphi_i \delta \\
\text{s.t.,} \forall i \in [0,\dots,N-1], \quad & \varphi_i \geq f_{\varphi,i}\left(m_i, P_{\mathrm{b},i}\right) \\
& m_i = m(k\delta) - \sum_{l=0}^{i-1} \varphi_l\, \delta \\
& E_i = E(k\delta) - \sum_{l=0}^{i-1} P_{\mathrm{b},l}\, \delta \\
& \underline{E} \leq E_i \leq \overline{E} \\
& \underline{\varphi}_i \leq \varphi_i \leq \overline{\varphi}_i \\
& \underline{P}_{\mathrm{b},i} \leq P_{\mathrm{b},i} \leq \overline{P}_{\mathrm{b},i}
\end{aligned} \tag{3.31}$$

where the bounds $\underline{\varphi}_i$, $\overline{\varphi}_i$, $\underline{P}_{\mathrm{b},i}$, $\overline{P}_{\mathrm{b},i}$ are given by

$$\overline{\varphi}_i = f_i(\overline{P}_{\mathrm{gt}}),$$

and, for $\mathcal{T} = \mathcal{P}$:

$$\begin{aligned}
\underline{\varphi}_i &= \max\Big\{f_i(\underline{P}_{\mathrm{gt}}), f_i(-\tfrac{\beta_1}{2\beta_2})\Big\} \\
\underline{P}_{\mathrm{b},i} &= \max\Big\{g_i\Big(h_i(\underline{P}_{\mathrm{em}})\Big), g_i\Big(h_i(-\tfrac{\kappa_{1,i}}{2\kappa_{2,i}})\Big)\Big\} \\
\overline{P}_{\mathrm{b},i} &= \min\Big\{g_i\Big(h_i(\overline{P}_{\mathrm{em}})\Big), g_i\Big(h_i(r_{\max,i})\Big)\Big\},
\end{aligned}$$

and, for $\mathcal{T} = \mathcal{S}$:

$$\begin{aligned}
\underline{\varphi}_i &= \max\Big\{f_i(\underline{P}_{\mathrm{gt}}), f_i(-\tfrac{\beta_1}{2\beta_2}), f_i\Big(h_{\mathrm{gen},i}(-\tfrac{\nu_1}{2\nu_2})\Big)\Big\} \\
\underline{P}_{\mathrm{b},i} &= \widetilde{f}_i(\underline{P}_{\mathrm{em}}, \overline{P}_{\mathrm{gt}}) \\
\overline{P}_{\mathrm{b},i} &= \min\Big\{\widetilde{f}_i(\overline{P}_{\mathrm{em}}, \underline{P}_{\mathrm{gt}}), \widetilde{f}_i(\overline{P}_{\mathrm{em}}, -\tfrac{\beta_1}{2\beta_2}), \widetilde{f}_i\Big(\overline{P}_{\mathrm{em}}, h_{\mathrm{gen},i}(-\tfrac{\nu_1}{2\nu_2})\Big), \tfrac{U^2}{2R}\Big\}.
\end{aligned}$$

3.5 Alternating direction method of multipliers

If $\underline{E} \leq E_i - \overline{P}_{\mathrm{b},i}\, \delta \leq \overline{E}\ \forall i$, so that at each time step there is enough energy in the battery to operate the electric motor at its maximum capacity, then the solution of (3.31) is given trivially by $P^*_{\mathrm{b},i} = \overline{P}_{\mathrm{b},i}$, $\forall i$, for both architectures. If this condition

is not satisfied, then an optimisation scheme is needed to solve problem (3.31). To make real-time implementation possible we propose a specialised alternating direction method of multipliers (ADMM) algorithm [50]. Problem (3.31) can be equivalently stated with inequality constraints appended to the objective function using indicator functions $\Lambda^x(x)$,

$$\begin{aligned}
\min_{\substack{\varphi, P_\mathrm{b}, m,\\ E, \chi, \xi, \zeta}} \quad & \sum_{i=0}^{N-1} \xi_i \delta + \Lambda^\chi(\chi_i) + \Lambda^E(E_i) + \Lambda^\varphi(\varphi_i) + \Lambda^{P_\mathrm{b}}(P_{\mathrm{b},i}) \\
\text{s.t.,} \forall i \in [0, \dots, N-1], \quad & \chi_i = \xi_i - f_{\varphi,i}(m_i, P_{\mathrm{b},i}) \\
& m_i = m(k\delta) - \sum_{l=0}^{i-1} \xi_l \delta \\
& E_i = E(k\delta) - \sum_{l=0}^{i-1} \zeta_l \delta \\
& \xi_i = \varphi_i \\
& \zeta_i = P_{\mathrm{b},i}
\end{aligned} \tag{3.32}$$

with $\underline{\chi} = 0$, $\overline{\chi} = \infty$, and

$$\Lambda^x(x) = \begin{cases} 0 & \text{if } \underline{x} \leq x \leq \overline{x}, \\ \infty & \text{otherwise.} \end{cases}$$

Note that we have introduced dummy variables ξ and ζ in order to simplify the solver iterations by separating variables.

We define an augmented Lagrangian function as

$$\begin{aligned}
& L(\chi, \xi, \zeta, E, P_\mathrm{b}, \varphi, m, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = \\
& \sum_{i=0}^{N-1} \Big(\xi_i \delta + \Lambda^\chi(\chi_i) + \Lambda^E(E_i) + \Lambda^\varphi(\varphi_i) + \Lambda^{P_\mathrm{b}}(P_{\mathrm{b},i}) \Big) \\
& + \frac{\sigma_1}{2} \sum_{i=0}^{N-1} \Big(\chi_i - \xi_i + f_{\varphi,i}(m_i, P_{\mathrm{b},i}) + \lambda_{1,i} \Big)^2 \\
& + \frac{\sigma_2}{2} \| m - m(k\delta)\Phi + \Psi\xi + \lambda_2 \|^2 \\
& + \frac{\sigma_3}{2} \| E - E(k\delta)\Phi + \Psi\zeta + \lambda_3 \|^2 \\
& + \frac{\sigma_4}{2} \| \xi - \varphi + \lambda_4 \|^2 \\
& + \frac{\sigma_5}{2} \| \zeta - P_\mathrm{b} + \lambda_5 \|^2,
\end{aligned}$$

where λ_i is a Lagrange multiplier and σ_i is a penalty parameter associated with the ith constraint, Φ is a vector of ones, and Ψ is the lower triangular matrix with lower triangular elements equal to δ.

Problem (3.32) can be rearranged in the canonical form

$$\begin{aligned} \min_{x,\,z} \quad & \hat{f}(x) + \hat{g}(z) \\ \text{s.t.} \quad & b(z) + Bx = c \end{aligned} \tag{3.33}$$

with

$$x = \begin{bmatrix} \chi^\top & \xi^\top & \zeta^\top & E^\top & \varphi^\top \end{bmatrix}^\top, \quad z = \begin{bmatrix} m^\top & P_b^\top \end{bmatrix}^\top, \quad \lambda = \begin{bmatrix} \lambda_1^\top & \lambda_2^\top & \lambda_3^\top & \lambda_4^\top & \lambda_5^\top \end{bmatrix}^\top,$$

$$\hat{f}(x) = \sum_{i=0}^{N-1} \xi_i \delta + \Lambda^\chi(\chi_i) + \Lambda^E(E_i) + \Lambda^\varphi(\varphi_i),$$

$$\hat{g}(z) = \sum_{i=0}^{N-1} \Lambda^{P_\mathrm{b}}(P_{\mathrm{b},i}),$$

$$B = \begin{bmatrix} I & -I & 0 & 0 & 0 \\ 0 & \Psi & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & \Psi & I & 0 \\ 0 & I & 0 & 0 & -I \end{bmatrix}, \quad b(z) = \begin{bmatrix} f_\varphi(m, P_b) \\ m \\ -P_b \\ 0 \\ 0 \end{bmatrix},$$

$$c = \begin{bmatrix} 0 & \Phi^\top m(k\delta) & 0 & \Phi^\top E(k\delta) & 0 \end{bmatrix}^\top.$$

We define the primal and dual residuals $r^{j+1} = b(z^{j+1}) + Bx^{j+1} - c$ and $s^{j+1} = [\nabla_z b(z^{j+1})]^\top R^j B(x^j - x^{j+1})$, where $R^j = \mathrm{diag}(\sigma_1^j I, \sigma_2^j I, \sigma_3^j I, \sigma_4^j I, \sigma_5^j I)$. Note that 0 and I are compatible zero and identity matrices. By comparison with [50], the present algorithm deals with a nonlinear $b(z)$ function in the equality constraint, which requires that the dual residual is defined in terms of the Jacobian $\nabla_z b$.

The ADMM iteration update is given by

$$\begin{aligned} \chi^{j+1} &= \pi^\chi(\xi^j - f_\varphi^j - \lambda_1^j), \\ \xi^{j+1} &= \Big((\sigma_1^j + \sigma_4^j) I + \sigma_2^j \Psi^\top \Psi\Big)^{-1} \Big[-\Phi\delta + \sigma_1^j(\chi^{j+1} + f_\varphi^j + \lambda_1^j) \\ &\qquad - \sigma_2^j \Psi^\top (m^j - m(k\delta)\Phi + \lambda_2^j) + \sigma_4^j \Big(\varphi^j - \lambda_4^j\Big) \Big], \\ \zeta^{j+1} &= (\sigma_5^j I + \sigma_3^j \Psi^\top \Psi)^{-1} \Big[-\sigma_3^j \Psi^\top \Big(E^j - E(k\delta)\Phi + \lambda_3^j\Big) + \sigma_5^j (P_\mathrm{b}^j - \lambda_5^j) \Big], \\ E^{j+1} &= \pi^E\Big(E(k\delta)\Phi - \Psi\zeta^{j+1} - \lambda_3^j\Big), \end{aligned}$$

$$
\begin{aligned}
P_{\mathrm{b},i}^{j+1} &= \pi^{P_\mathrm{b}}\Big(\arg\min_{P_{\mathrm{b},i}}\Big\{\frac{\sigma_1^j}{2}\Big[\chi_i^{j+1}-\xi_i^{j+1}+f_{\varphi,i}(m_i^j,P_{\mathrm{b},i})+\lambda_{1,i}^j\Big]^2 \\
&\qquad\qquad +\frac{\sigma_5^j}{2}\Big[\zeta_i^{j+1}-P_{\mathrm{b},i}+\lambda_{5,i}^j\Big]^2\Big\}\Big), \\
\varphi^{j+1} &= \pi^{\varphi}\Big(\xi^{j+1}+\lambda_4^j\Big), \\
m_i^{j+1} &= \arg\min_{m_i}\Big\{\frac{\sigma_1^j}{2}\Big[\chi_i^{j+1}-\xi_i^{j+1}+f_{\varphi,i}(m_i,P_{\mathrm{b},i}^{j+1})+\lambda_{1,i}^j\Big]^2 \\
&\qquad\qquad +\frac{\sigma_2^j}{2}\Big[m_i-m(k\delta)\Phi+[\Psi\xi^{j+1}]_i+\lambda_2^j\Big]^2\Big\}, \\
\lambda_1^{j+1} &= \lambda_1^j+\chi^{j+1}-\xi^{j+1}+f_\varphi^{j+1}, \\
\lambda_2^{j+1} &= \lambda_2^j+m^{j+1}-m(k\delta)\Phi+\Psi\xi^{j+1}, \\
\lambda_3^{j+1} &= \lambda_3^j+E^{j+1}-E(k\delta)\Phi+\Psi\zeta^{j+1}, \\
\lambda_4^{j+1} &= \lambda_4^j+\xi^{j+1}-\varphi^{j+1}, \\
\lambda_5^{j+1} &= \lambda_5^j+\zeta^{j+1}-P_\mathrm{b}^{j+1},
\end{aligned}
$$

where $f_\varphi^j=[f_{\varphi,0}^j\ \cdots\ f_{\varphi,N-1}^j]^\top$ and $\pi^x(y)$ denotes the projection $\max\{\min\{y,\overline{x}\},\underline{x}\}$. The penalty parameters σ_n^j, $n=1,2,3,4,5$ are updated at intervals of F_σ iterations (provided $10<\max\Big\{\frac{||r^{j+1}||}{\max\{||b(z^{j+1})||,||Bx^{j+1}||,||c||\}},\frac{||s^{j+1}||}{||\nabla_z b(z^{j+1})^\top\lambda^{j+1}||}\Big\}$) according to the rule

$$
\begin{aligned}
\tau^{j+1} &= \begin{cases}\Gamma & \text{if } 1\le\Gamma<\tau_{\max},\\ \Gamma^{-1} & \text{if } \tau_{\max}^{-1}<\Gamma<1,\\ \tau_{\max} & \text{otherwise,}\end{cases} \\
\sigma_n^{j+1} &= \begin{cases}\sigma_n^j\tau^{j+1} & \text{if } \|r_n^{j+1}\|>\mu\,\|s^{j+1}\|,\\ \sigma_n^j/\tau^{j+1} & \text{if } \|s^{j+1}\|>\mu\,\|r_n^{j+1}\|,\\ \sigma_n^j & \text{otherwise,}\end{cases} \\
R^{j+1} &= \mathrm{diag}(\sigma_1^{j+1}I,\sigma_2^{j+1}I,\sigma_3^{j+1}I,\sigma_4^{j+1}I,\sigma_5^{j+1}I),
\end{aligned}
$$

where $\Gamma=\sqrt{\|r^{j+1}\|/\|s^{j+1}\|}$ and r_n^{j+1} denotes the rows of r^{j+1} associated with the nth constraint, $1\le n\le 5$.

The updates for χ, E and the multipliers λ_1, λ_2, λ_3, λ_4, λ_5 involve only vector additions, summations and projections. The equations $\Big((\sigma_1^j+\sigma_4^j)I+\sigma_3^j\Psi^\top\Psi\Big)\xi=c_1$ and $(\sigma_5^jI+\sigma_3^j\Psi^\top\Psi)\zeta=c_2$ can be solved for ξ and ζ (for given c_1 and c_2) in $O(N)$ operations using appropriate Cholesky factorisations (for details see [31], Prop. 3). The Cholesky factors can be reused until the penalty parameters σ_n^j are updated,

otherwise the updates for ξ and ζ require only scalar multiplications and vector summations. The updates for P_b and m require minimisation of scalar convex functions and can be performed using Newton's method.

The algorithm is initialised with

$$P_b^0 = \Phi\overline{P}_b, \quad \zeta^0 = P_b^0, \quad \xi^0 = \underline{\varphi}, \quad \varphi^0 = \xi^0,$$
$$E^0 = \pi^E\left(\Phi E(k\delta) - \Psi\zeta^0\right), \quad m^0 = \Phi m(k\delta) - \Psi\xi^0$$
$$\chi^0 = \pi^\chi(\xi^0 - f_\varphi^0), \quad \lambda_1^0 = \lambda_2^0 = \lambda_3^0 = \lambda_4^0 = \lambda_5^0 = 0\Phi$$
$$R^0 = \mathrm{diag}(50I, 3.69\times10^{-7}I, 6.96\times10^{-7}I, 20.29I, 0.83I),$$

and stopped when $\|r^{j+1}\| \le \epsilon_P$ and $\|s^{j+1}\| \le \epsilon_D$ or $j > 10^5$, where, following [50],

$$\epsilon_P = \sqrt{5N}\epsilon_{\text{abs}} + \epsilon_{\text{rel}}\max\{||b(z^{j+1})||, ||Bx^{j+1}||, ||c||\},$$
$$\epsilon_D = \sqrt{2N}\epsilon_{\text{abs}} + \epsilon_{\text{rel}}||[\nabla_z b(z^{j+1})]^\top\lambda^{j+1}||.$$

The penalty parameters σ_n^j are initialised so that all terms of the Lagrangian are initially of the same order of magnitude.

3.6 Numerical results

In this section we introduce an energy management case study involving a representative hybrid-electric passenger aircraft and solve optimisation problem (3.31) within this context using the ADMM algorithm as presented in section 3.5. The simulation results are analysed and the performance of the algorithm is discussed in terms of its computational requirements and robustness to variations in model parameters.

3.6.1 Simulation scenario

The parameters of the model used in simulations are shown in Table 3.1. These are based on published data for the BAe 146 aircraft. The conventional BAe 146 propulsion system is replaced by hybrid-electric propulsion systems[2] in either parallel or series configuration (as illustrated in Figs. 3.2 and 3.3), both of which

[2]In order to maintain a constant maximum take-off weight (MTOW), the excess mass from the batteries, electric motors, generators and electrical distribution systems can be compensated by cuts in passenger count and fuel mass.

were equipped with the same battery size. The aircraft is powered by a combination of 4 such systems.

Table 3.1: Model parameters

Parameter	Symbol	Value	Units
Mass (MTOW)	m	42000	kg
Gravity acceleration	g	9.81	$\mathrm{m\,s^{-2}}$
Wing area	S	77.3	$\mathrm{m^2}$
Lift coefficients	b_0	0.43	–
	b_1	0.11	$\mathrm{deg^{-1}}$
Drag coefficients	a_0	0.029	–
	a_1	0.004	$\mathrm{deg^{-1}}$
	a_2	5.3e−4	$\mathrm{deg^{-2}}$
Angle of attack range	$[\underline{\alpha};\overline{\alpha}]$	$[-3.9; 10]$	deg
# of propulsion systems	n	4	–
Total fuel mass	m_{fuel}	$1000 \times n$	kg
Total battery mass	m_{b}	$2000 \times n$	kg
Battery energy density	e_{b}	0.875	$\mathrm{MJ\,kg^{-1}}$
Fuel map coefficients	β_0	0.0327	$\mathrm{kg\,s^{-1}}$
	β_1	0.0821	$\mathrm{kg\,MJ^{-1}}$
Generator coefficients	ν_0	0.08	MW
	ν_1	1	–
Total battery SOC range	$\left[\underline{E};\overline{E}\right] \times n$	$[350; 1487] \times n$	MJ
Gas turbine power range	$\left[\underline{P}_{\mathrm{gt}};\overline{P}_{\mathrm{gt}}\right]$	$[0; 5]$	MW
Motor power range	$\left[\underline{P}_{\mathrm{em}};\overline{P}_{\mathrm{em}}\right]$	$[0; 5]$	MW
Battery o.c. voltage	U	1500	V
Battery resistance	R	0.035	ohm
Flight time	T	3600	s

For the purposes of this study it is assumed that velocity and height profiles are known a priori as a result of a fixed flight plan determined prior to take-off. We consider an exemplary 1-hour flight at a true airspeed (TAS) of 190 m/s for a typical 100-seat passenger aircraft. The flight path (height and velocity profile) is shown in Figure 3.4.

The electric loss map coefficients $\kappa_{2,i}, \kappa_{1,i}, \kappa_{0,i}$ can be estimated $\forall i$ from these profiles. First, the drive power P_{drv} is approximated a priori, e.g. by assuming a conventional gas-turbine-powered flight. Then, the fan shaft rotation speed, $\omega_{\mathrm{drv},i}$ (equal to the electric motor shaft rotation speed in both configurations), is

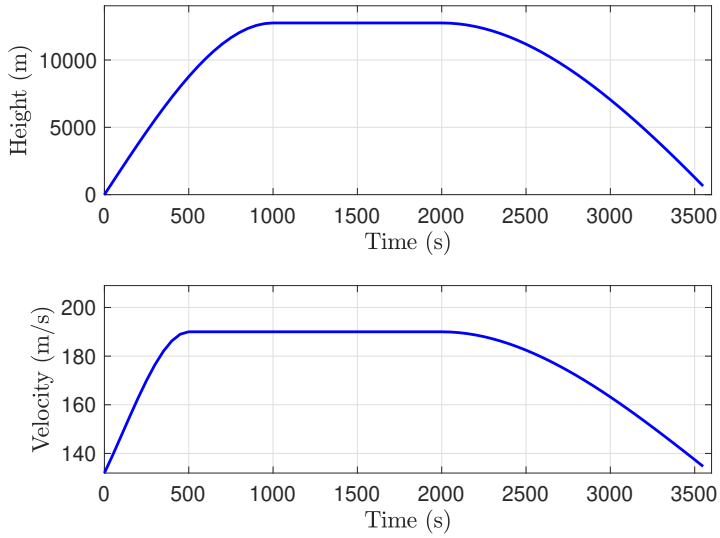


Figure 3.4: Height and velocity profiles for the mission.

interpolated from a precomputed look-up table relating measured shaft rotation speed, altitude and drive power at a given Mach number. For example, a Mach number of 0.55 (190 m/s TAS) gives the relationship shown in Fig. 3.5, which was obtained by scaling a proprietary fan design for the thrust range of the BAe 146 aircraft. The non-dimensional rotation speed Ω is thus estimated at a given altitude, Mach number and drive power using the map in Fig. 3.5, and the shaft rotation speed is inferred from

$$\omega_{drv} = \frac{156.7}{100}\frac{\pi}{30}\Omega\sqrt{T_{in}},$$

where $T_{in} = T_0(h)+v^2/2c_p$ is the temperature at inlet of the fan, $c_p = 1005\ \mathrm{JK^{-1}kg^{-1}}$ is the specific heat of air at constant pressure and $T_0(h)$ is the temperature of air at altitude h. Finally, the coefficients are interpolated from a precomputed record of losses in the electric motor as a function of ω_{drv}.

The gas turbine fuel map and generator loss map used in this study are approximately linear ($\beta_2 \approx 0$ and $\nu_2 \approx 0$) for the range of power conditions considered[3], and the coefficients are constant as discussed in Section 3.4.

[3]Note that under this assumption the function $f_{\varphi,i}$ defined in equation (3.27) would be quadratic since $P_{\mathrm{drv},i}$ and g_i^{-1} are quadratic functions. This implies that problem (3.31) could be reformulated as a second-order cone program and solved efficiently with interior-points methods.

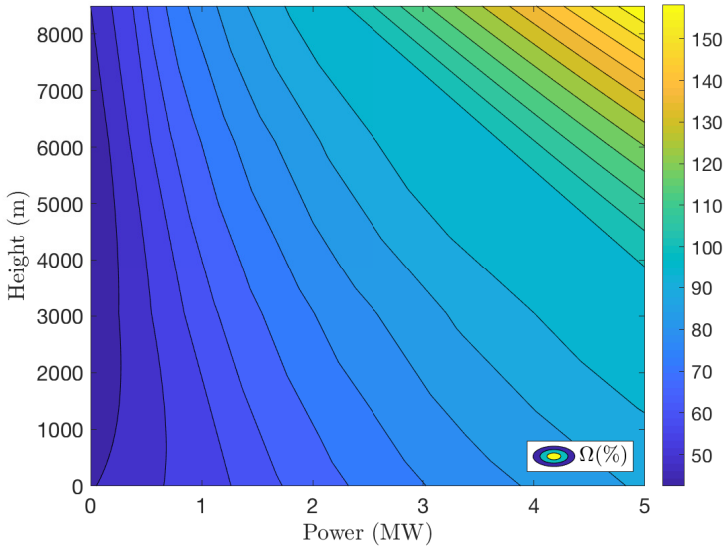


Figure 3.5: Contour plot relating drive power, altitude and non-dimensional rotation speed for a Mach number of 0.55.

3.6.2 Results

The mission is simulated in both configurations with sampling interval $\delta = 60$ s over a one-hour shrinking horizon by solving the optimisation problem (3.31) at each time step and implementing the first element of the optimal power split sequence as an MPC law. The tolerance is set to $\epsilon_{\text{rel}} = 5\text{e}{-6}$, $\epsilon_{\text{abs}} = 0$ and the penalty parameters are updated at intervals of $F_\sigma = 500$ iterations.

The closed-loop ADMM solution to the energy management control strategy is shown in Fig. 3.6 for both parallel and series configurations. The solutions are for a single propulsion system (so all quantities should be multiplied by $n = 4$ to obtain the results for the whole aircraft). The plots represent the evolution of the relevant power terms in the power balance equations (3.13) and (3.14): P_{drv}, P_{em}, P_{gt} for the parallel configuration and P_{el}, P_{c}, P_{gen} for the corresponding terms in the series configuration. It should be noted that the solution is similar in both configurations. A striking feature of the solutions is the tendency to allocate more electrical power at the end of the flight. An intuitive explanation for this phenomenon is that the fuel burnt by using the gas turbine at the beginning of the flight reduces the mass of the aircraft, consequently reducing the power required

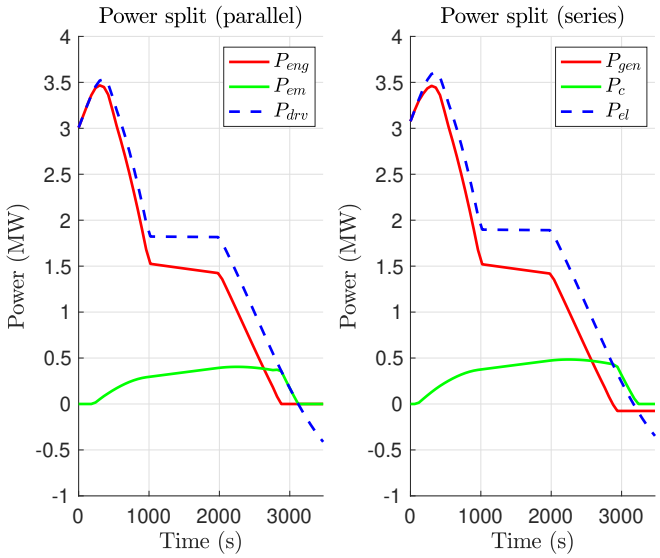


Figure 3.6: Closed-loop ADMM solution to the energy management problem in the parallel and series configurations, shown for 1 system (4 overall).

to be produced by the fan later in the flight. This effect is amplified if the rate of fuel consumption is increased, as seen in Fig. 3.7 comparing the electrical power profiles with different fuel consumption coefficients (β).

It should be noted that a concurrent effect arises from the losses in the battery electric bus. The nonlinear loss map g between the battery chemical power P_{b} and effective power P_{c} tends to penalise large electrical power peaks thus flattening the electrical power distribution. This is seen in Fig. 3.8 which shows the electrical power profiles with different values for the battery equivalent circuit resistance: for smaller resistances the electrical losses at high power outputs is reduced so the power profile shows greater variation over time.

Fig. 3.9 compares the evolution of battery SOC and fuel consumption for both configurations. As illustrated, the series propulsion architecture consumes slightly more fuel because it implements one more electric machine with associated losses, and so the electrical power is larger for a given flight profile. The selection of a particular configuration is thus motivated by a trade-off between efficiency and complexity of aero-mechanical integration.

The distribution of the electrical power over time is also illustrated in Fig. 3.10 in comparison with other energy management strategies. The charge-depleting-charge-

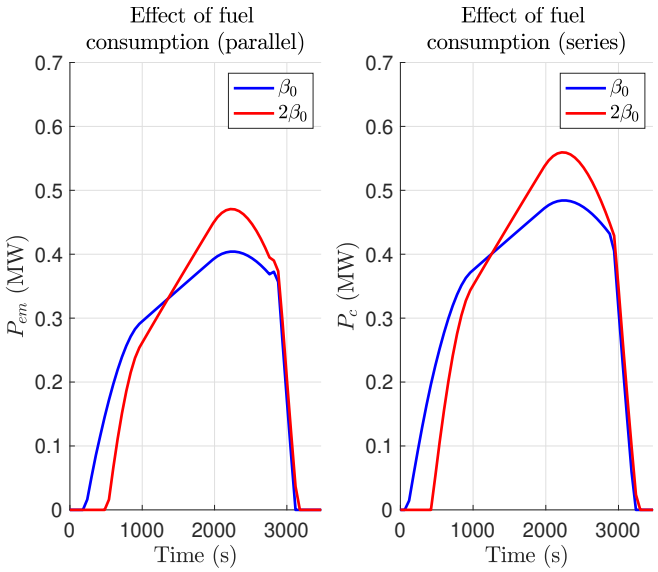


Figure 3.7: Effect of changing fuel map coefficient β_0 (single system).

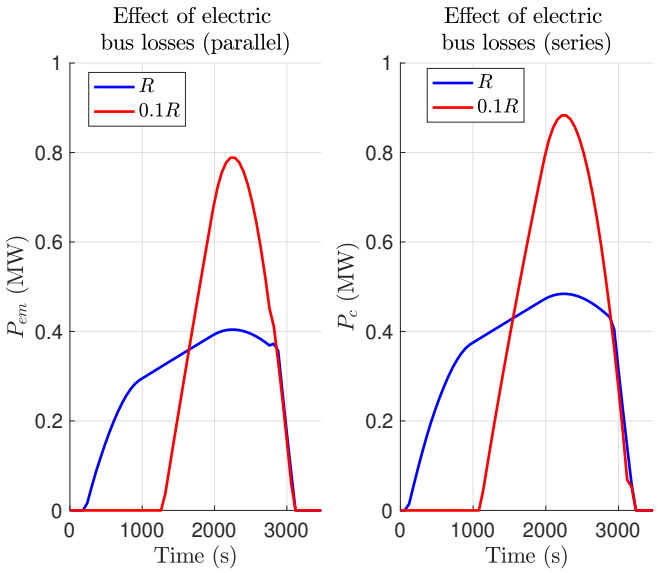


Figure 3.8: Effect of changing battery resistance R (single system).

sustaining (CDCS) strategy is a heuristic that uses all the electrical energy at the beginning of the flight until the battery is depleted and then relies solely on the gas turbine for the remainder of the flight. Interestingly, the proposed ADMM-based approach is the antithesis of this strategy, allocating a non-negligible part of the electrical power at the end of the flight. The third strategy illustrated in Fig. 3.10 uses the ADMM algorithm but ignores the aircraft mass variation. Interestingly, this (necessarily suboptimal) solution distributes the electrical power uniformly

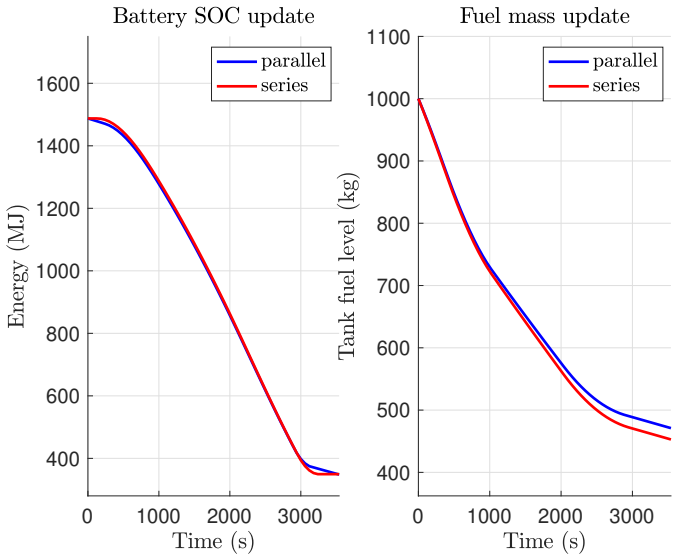


Figure 3.9: Comparison of battery and fuel consumption in the parallel and series configurations, shown for 1 system.

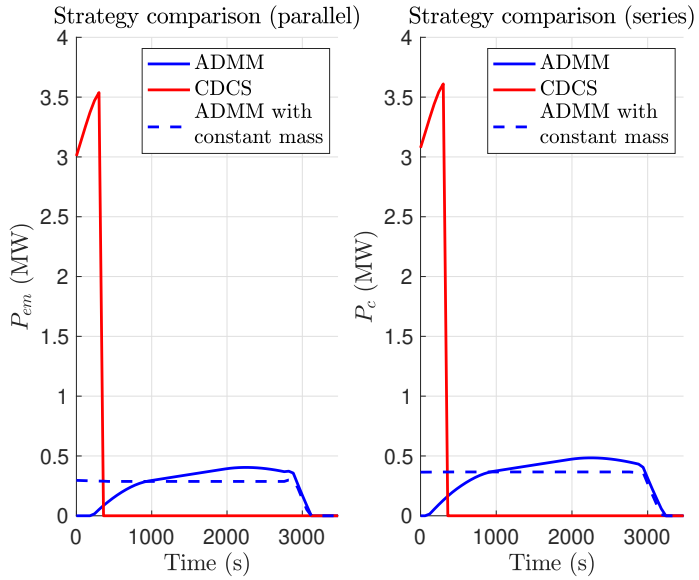


Figure 3.10: Comparison of ADMM, CDCS and ADMM with constant mass (parallel and series architectures, single system).

over the duration of the flight. In this case the strategy is dominated by the need to reduce electrical losses; neglecting the aircraft mass variation means that the potential savings due to fuel burn early in the flight are not exploited.

The superiority of the presented variable mass ADMM solver over other strategies is shown in Table 3.2. The heuristic CDCS strategy is used as a benchmark case. It is shown that the fuel savings with the mass-varying ADMM solver are

Table 3.2: Fuel comparison

	Configuration			
	Parallel		Series	
Method	**Fuel (kg)**	**Saving (%)**	**Fuel (kg)**	**Saving (%)**
CDCS	2152	–	2231	–
Constant mass	2123	1.3	2192	1.7
Variable mass	2115	1.7	2188	1.9

superior to other strategies, in both parallel and series configuration. In the parallel configuration, the proposed energy management strategy achieves a fuel consumption of 2115 kg, namely 1.7% less than with CDCS. Likewise, in the series configuration, a fuel consumption of 2188 kg is reported, which corresponds to a 1.9% decrease over CDCS.

As expected, the series architecture consumes more fuel than the parallel architecture. This is because series propulsion architectures employ an additional electrical machine and thus consume more electrical power due to inherent losses. Despite being less efficient, the series architecture has potential advantages in multi-propulsor configurations and is less mechanically complex than the parallel configuration.

It should be noted that the same aircraft equipped with a conventional gas turbine propulsion system would burn $F_{\mathrm{gt}} = 2403$ kg over the same scenario flight using the same models of powertrain components. However, in practice the conventional powertrain would be lighter since aviation fuels have a much higher energy density than batteries, so a direct comparison of fuel consumption is not possible.

Finally, we consider extensions of this case study to demonstrate the full potential of the proposed solver. We consider the same flight scenario but now assume that 1) the maximum gas turbine power is $\overline{P}_{\mathrm{gt}} = 3$ MW, and 2) the propulsion unit is capable of converting negative drive power during descent into electricity to recharge the battery (i.e. "windmilling"). This operation mode can be enforced by assuming a recovery efficiency η_w and setting $\underline{P}_{\mathrm{b}} = \overline{P}_{\mathrm{b}} = g\Big(h(\eta_w P_{\mathrm{drv},i})\Big)$ for all time steps i such that $P_{\mathrm{drv},i} < 0$. Alternatively, the objective in problems (3.15) and (3.16) can be modified to introduce a terminal term $-\lambda E_N$ so as to maximise the SOC of the battery at the end of the flight. The coefficient λ should be small (typically

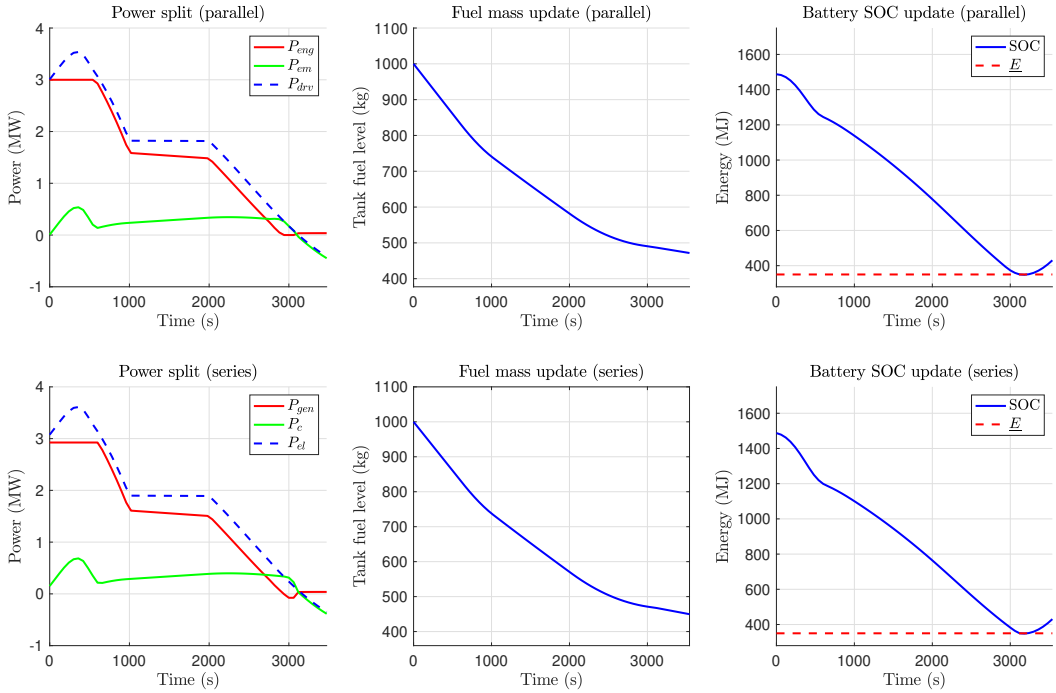


Figure 3.11: Effect of windmilling and gas turbine saturation (single system).

$\lambda = 0.1$) to avoid affecting the adverse objective of minimising fuel consumption. This could be detrimental in scenarios when the battery cannot be charged fully during descent, resulting in a suboptimal solution with excess SOC before the descent that could have otherwise been traded for fuel.

Figure 3.11 shows the impact of these modifications. Gas turbine saturation causes more electrical energy to be allocated to the point at which the gas turbine saturates. The potential for energy recovery via a windmilling mode is apparent at the end of the flight, where the drive power is negative, and the battery SOC increases during this part of the descent. It has been assumed for simplicity that the recovery process is ideal, that the electric motor can be operated as a generator and that the fan can be operated in reverse (requiring e.g. a variable pitch fan). In practice, we would expect recovery efficiencies between $10\% - 20\%$ with current technology.

3.6.3 Solver performance

We next consider the convergence and robustness properties of the proposed ADMM solver. Instead of solving the optimisation problem at successive time steps to

derive the MPC law (as in Section 3.6.2), we consider solving only one instance of the optimisation problem in order to simplify the analysis. We show that the proposed solver is robust to changes in the flight profile, aircraft parameters and problem dimension. The parallel configuration is considered here, all results being qualitatively equivalent for the series configuration.

Section 3.6.2 assumed fixed values for parameters that influence convergence rate (tolerances, sample rate and penalty parameter update frequency). To investigate the effect of changing tolerances, Fig. 3.12 shows accuracy relative to the optimal solution (obtained by solving the problem with optimisation package CVX [123] and solver SDPT3 [124] and comparing total fuel consumption), number of iterations to completion, and computation time as a function of relative tolerance ϵ_{rel}. The latter was varied while keeping other parameters constant (with $\epsilon_{\text{abs}} = 0$, $F_\sigma = 10^5$, $\delta = 10s$). As expected, as tolerance decreases, accuracy increases at the expense of a larger number of iterations and a consequent increase in computation.

It is possible to reduce the tolerance without incurring additional computational cost if the ADMM algorithm is augmented with a penalty parameter update scheme as introduced in section 3.5. This is illustrated in Fig. 3.13, which was obtained by varying the update frequency $1/F_\sigma$ while keeping other parameters constant (with $\epsilon_{\text{abs}} = 0$, $\epsilon_{\text{rel}} = 5 \times 10^{-5}$, $\delta = 10s$). The number of iterations required (and consequently the computation time) decreases as the update frequency increases. However, this tends to decrease accuracy with respect to the CVX solution. The frequency update should thus be selected with care so as not to affect accuracy.

The influence of the sampling interval δ on computation time is shown in Fig. 3.14 by varying the problem dimension ($N = T/\delta$), with all other parameters kept constant ($\epsilon_{\text{abs}} = 0$, $\epsilon_{\text{rel}} = 5 \times 10^{-5}$, $F_\sigma = 50$). Computation time increases as problem dimension increases, however, the proposed ADMM solver provides significant computation time reduction relative to CVX, allowing longer prediction horizons and better real-time convergence.

Experiments were performed to compare the proposed ADMM algorithm for the convex problem (3.31) with direct solution of the nonconvex problem (3.15).

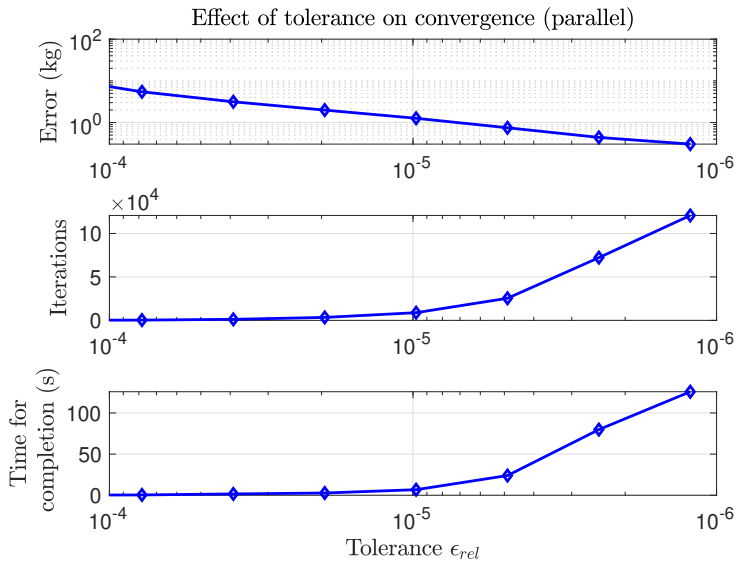


Figure 3.12: Effect of varying the relative tolerance on ADMM convergence.

Retaining only assumption 1) from section 3.4, the nonconvex problem was solved using a general purpose nonlinear programming solver (fmincon [125]) with $\delta = 60\,\mathrm{s}$, which converged within 82 s. Under the same conditions ADMM (implemented in Matlab) converged within 0.5 s. To compare fmincon and ADMM solutions, a Monte Carlo simulation was conducted by solving 100 problem instances with battery size randomly sampled from a uniform distribution. For each scenario the mean absolute error between the solutions (P_b) of both solvers was computed. The variance of the error distribution is 9.3×10^{-5} MW^2, showing good agreement between fmincon and ADMM. Future work could investigate comparisons with state-of-the-art nonlinear programming solvers such as IPOPT [126] that might show superior performance over fmincon.

Finally, robustness to changes in the mission parameters is investigated in Figures 3.15-3.17 where CVX and ADMM solutions are compared for modified simulation scenarios and fixed convergence parameters ($\epsilon_{\text{abs}} = 0$, $\epsilon_{\text{rel}} = 5 \times 10^{-5}$, $F_\sigma = 50$, $\delta = 10s$). These results show that the ADMM solution matches the solution obtained using CVX, thus demonstrating robustness to changes in problem-specific parameters.

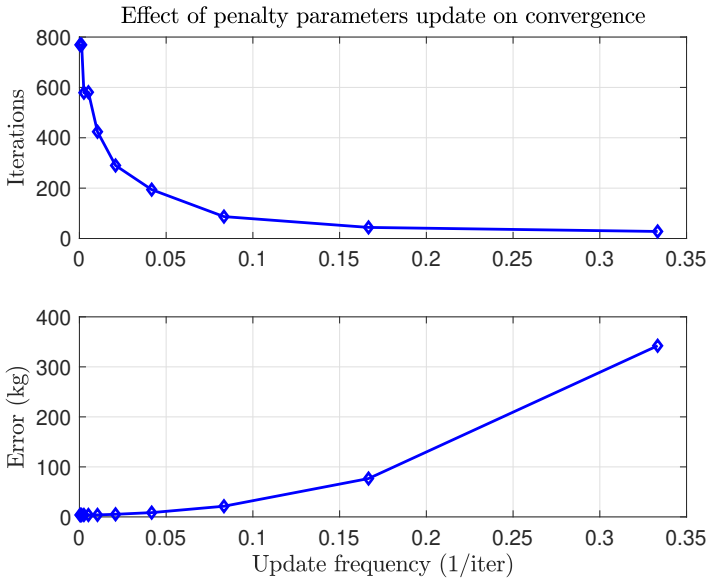


Figure 3.13: Effect of varying the penalty parameters update frequency on ADMM convergence.

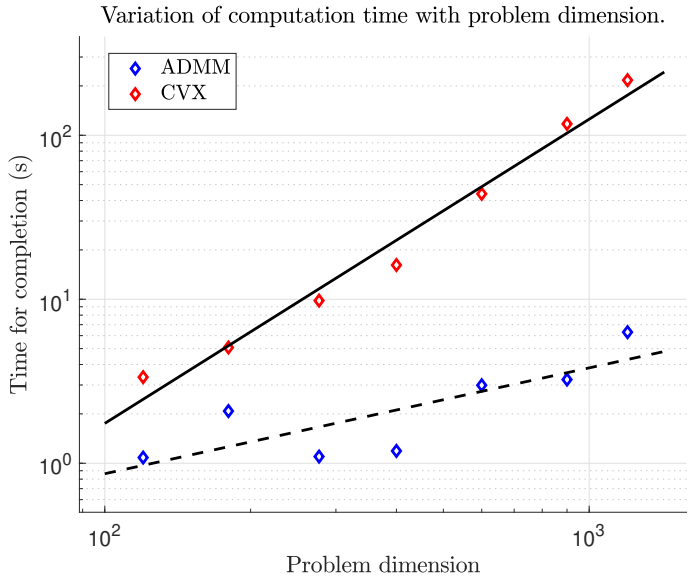


Figure 3.14: Effect of problem dimension on computation time.

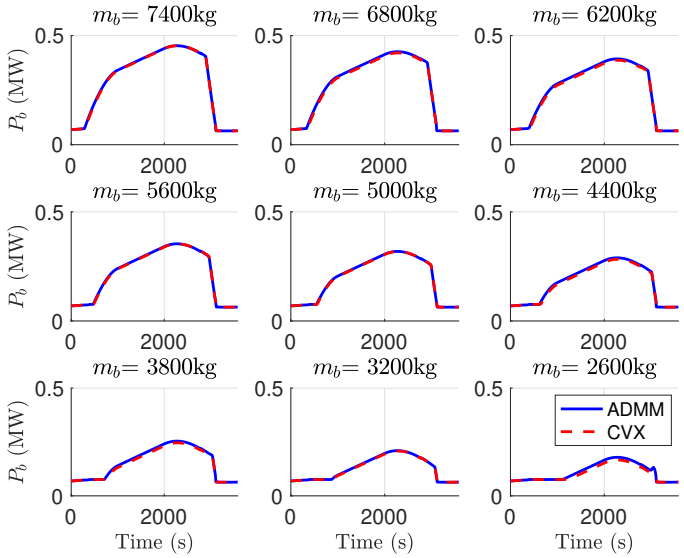


Figure 3.15: ADMM and CVX solutions for various battery masses.

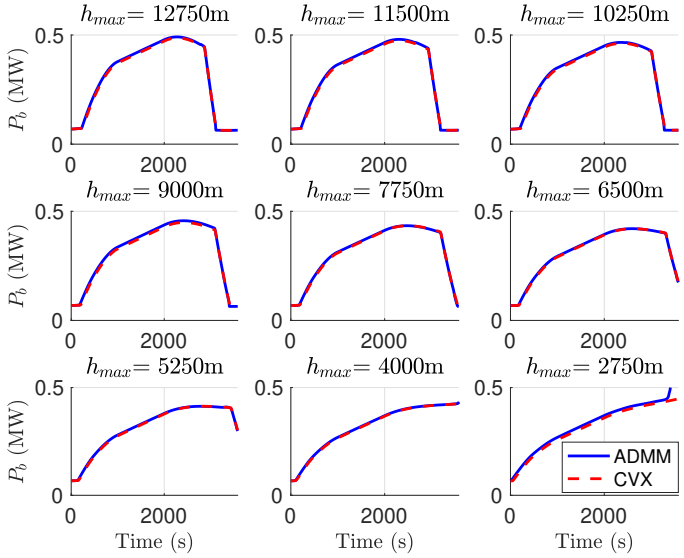


Figure 3.16: ADMM and CVX solutions for various maximum altitudes.

3.7 Conclusions

This chapter presents a fast and robust ADMM algorithm to solve the energy management problem for a hybrid electric aircraft in parallel and series configurations. A convex program is derived from the associated optimisation problem and the high degree of separability in the optimisation variables is exploited in the design of the solver. The ADMM solver was shown to produce similar results to the general

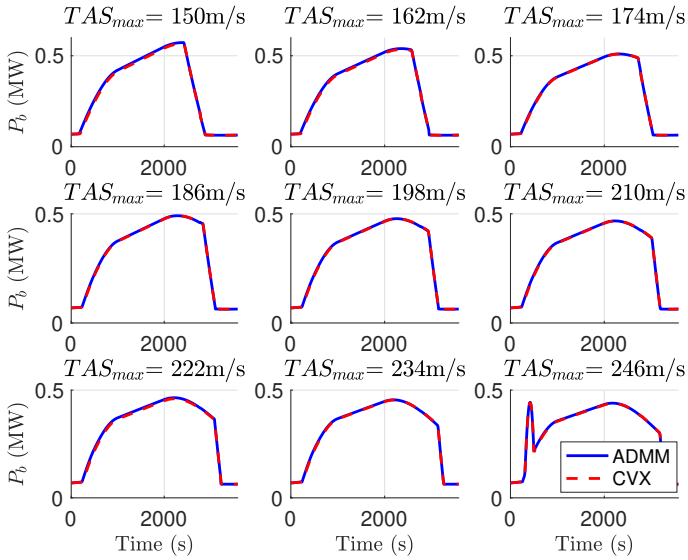


Figure 3.17: ADMM and CVX solutions for various maximum TAS.

purpose convex optimisation package CVX (with solver SDPT3) for a wide range of scenarios, while significantly outperforming CVX in terms of computation times. Significant fuel savings were achieved by comparison to heuristic strategies.

An extension of the proposed approach could optimise gas turbine speed given an estimate of its power output (possibly within an iterative scheme), removing the need for the assumption on gas turbine speed for the series configuration.

We note as well that the principles developed here for energy management could be applied to the problem of optimal design and sizing of powertrain components.

Another extension would be to investigate robustness of the proposed approach to power demand disturbances. Although the flight path is fixed and the aircraft flight dynamics are prescribed in the MPC optimisation, the predicted power demand is likely to be inexact and this would introduce disturbance terms into the dynamics of battery SOC and fuel mass.

Future work will also investigate the application of the proposed algorithm to solve the energy management problem for other types of hybrid vehicles (e.g. hybrid-electric vertical take-off and landing (VTOL) aircraft with applications to urban air mobility). Although we do not consider the problem of energy management

specifically for VTOL aircraft in this thesis, the next chapter is concerned with trajectory optimisation of VTOL aircraft using convex optimisation.

4

Robust trajectory optimisation of VTOL aircraft

Contents

4.1 Introduction

Urban air mobility (UAM) is set to revolutionise transportation of people and goods in congested cities while reducing the pressure on urban traffic [55]. A key enabler of this revolution is a new class of vertical take-off and landing (VTOL) aircraft that can be operated autonomously in constrained environments. Capable of transitions between vertical and forward flight, VTOL aircraft are among the most

promising technologies identified for operation in narrow space while guaranteeing the endurance of wing-borne flight. To deploy these flying taxis safely in cities, the development of reliable control methods and optimisation algorithms is a vital requirement.

The most critical phase of flight for VTOL aircraft is the transition between powered lift and wing-borne flight. Achieving successful transitions requires computing feasible trajectories which is a difficult nonlinear program (NLP) as it involves nonlinear flight dynamics.

In this chapter, we investigate a computationally tractable robust optimisation algorithm for the trajectory optimisation of VTOL aircraft. Although we consider here the problem of tiltwing aircraft transition, the method described is equally applicable to tiltrotors and other forms of VTOL aircraft in UAM scenarios.

Our approach is based on a DC decomposition of the nonlinear dynamics, followed by successive linearisations around predicted trajectories. At each iteration of the scheme, the linearisation error is treated as a bounded disturbance in a robust optimisation framework. Due to the DC form of the dynamics, this error is necessarily convex and can be bounded tightly since it takes its maximum at the boundary of the domain. Consequently, the trajectories of model states can be bounded by a set of convex inequalities. These inequalities form the basis of a computationally-tractable convex optimisation based on successive linearisations for the trajectory generation of VTOL aircraft. Finally, because DC decompositions can always be obtained for dynamical systems with sufficient smoothness[1], the method can be generalised to a broad class of systems, extending considerably the scope of our approach.

The contribution of this chapter is twofold: i) we solve an open problem in trajectory optimisation of VTOL aircraft by allowing transitions at near constant altitude and high angle of attack while guaranteeing safety and computational tractability of the scheme; ii) we introduce a new robust optimisation method for

[1]Following a classical result in [127], we require that the dynamics f belongs to the class $\mathcal{C}^2$.

nonlinear systems based on DC decomposition and successive linearisation of the dynamics and demonstrate the applicability of the procedure to a realistic case study.

This chapter is organised as follows. After discussing related work from the literature in Section 4.2, we develop a mathematical model of a tiltwing VTOL aircraft in Section 4.3. In Section 4.4, we formulate the trajectory optimisation problem and discuss a series of simplifications to obtain a convex program, leveraging ideas from DC decomposition and robust optimisation. Section 4.5 discusses simulation results obtained for a UAM case study based on the Airbus A^3 Vahana. Section 4.6 presents conclusions.

4.2 Related work

Our approach is related to algorithms based on successive linearisations [83] that treat the linearisation error as a bounded disturbance, although the novelty here is to use the curvature information from the dynamics to bound the linearisation error tightly. A parallel can also be made with DC programming and the convex-concave procedure [128] where the nonconvex part of the dynamics is linearised around successive guess trajectories, generating a sequence of convex programs. Contrary to the convex-concave procedure, our approach includes the linearisation error in the optimisation and considers nonlinear equality constraints typical of dynamical systems. Another related approach can be found in [129] where nonlinear convex equality constraints $h(x) = 0$ in optimisation problems are relaxed and approximated by a set of convex and linear inequalities

$$\begin{aligned} h(x) - \varepsilon &\leq 0, \\ -h(x^\circ) - \nabla h^\top(x^\circ)(x - x^\circ) &\leq 0, \\ -\varepsilon &\leq 0, \end{aligned}$$

where ε accounts for the linearisation error and is also added to the objective as a penalty term. The drawback of this approach is that, although it is shown to converge for certain values of the coefficient of the penalty term, there are no guarantees of convergence in general. Moreover, contrary to our present method, the approach

in [129] does not guarantee feasibility with respect to the original optimisation problem in case of early termination of the algorithm when $\varepsilon \neq 0$ at termination. Finally our approach is different in that it relies on a reparameterisation of the state in terms of uncertainty sets, and can thus be reformulated as a multi-stage optimal control problem in the framework of tube MPC [69], see Chapter 5.

4.3 Modeling

Consider a longitudinal point-mass model of a tiltwing VTOL aircraft equipped with propellers as shown in Figure 4.1. The equations of motion (EOM) are given in polar form by [68, 120]

$$m\dot{V} = T\cos\alpha - D - mg\sin\gamma, \quad V(t_0) = V_0, \tag{4.1}$$

$$mV\dot{\gamma} = T\sin\alpha + L - mg\cos\gamma, \quad \gamma(t_0) = \gamma_0, \tag{4.2}$$

$$J_w\ddot{i}_w = M, \quad i_w(t_0) = i_0, \quad \dot{i}_w(t_0) = \Omega_0, \tag{4.3}$$

$$\dot{X} = V\cos\gamma, \qquad \dot{Z} = -V\sin\gamma, \tag{4.4}$$

where the control inputs are the thrust magnitude T and the total torque M delivered by the tilting actuators, and the model states are the aircraft velocity magnitude V, the flight path angle γ (defined as the angle of the velocity vector from horizontal), the tiltwing angle i_w and its derivative $\dot{i}_w$, and the position (X, Z) with respect to inertial frame O_{XZ}. Additional variables are the lift force L, drag force D and the angle of attack α.

The following input and state constraints apply

$$i_w + \theta = \alpha + \gamma, \; \underline{M} \leq M \leq \overline{M}, \tag{4.5}$$

$$0 \leq T \leq \overline{T}, \quad 0 \leq V \leq \overline{V}, \tag{4.6}$$

$$V(t_0) = V_0 \text{ and } V(t_f) = V_f, \tag{4.7}$$

$$\underline{a} \leq \dot{V} \leq \overline{a}. \tag{4.8}$$

Here θ is the pitch angle, defined as the angle of the fuselage axis from horizontal. For passenger comfort, θ is regulated via the elevator to track a constant reference $\theta^* = 0$.

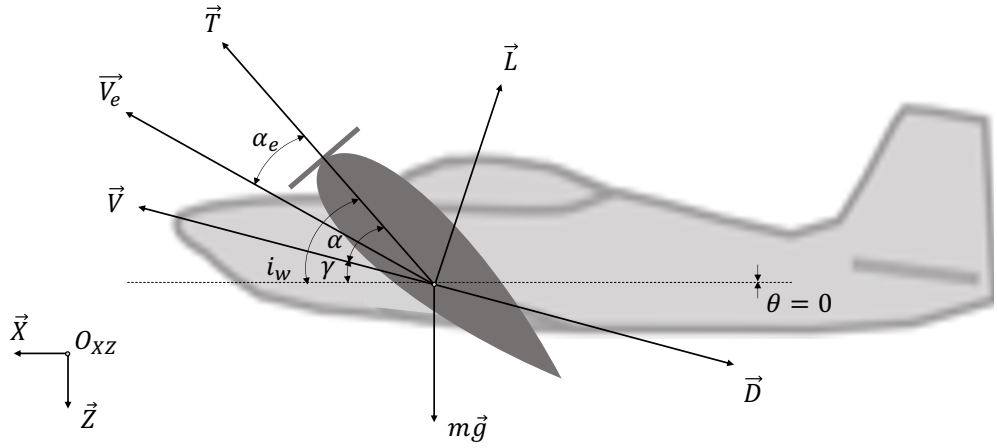


Figure 4.1: Force and velocity definitions for a VTOL aircraft

In order to account for the effect of the propeller wake on the wing, the flow velocity downstream is augmented by the induced velocity of the propeller. From momentum theory [61, 68], the effective velocity V_e and effective angle of attack α_e seen by the wing are given by

$$\alpha_e = \arcsin\left(\frac{V}{V_e}\sin\alpha\right), \tag{4.9}$$

$$V_e = \sqrt{V^2 + \frac{2T}{\rho A n}}. \tag{4.10}$$

where ρ is the air density, A the rotor disk area, n the number of propellers.

Assuming that the wing is fully immersed in the wake, and that $\alpha_e \ll 1$ to avoid wing stall (this will be verified *a posteriori* from simulation results), the lift and drag are modeled as follows [68]

$$D = \tfrac{1}{2}\rho S(a_2\alpha_e^2 + a_1\alpha_e + a_0)V_e^2 \approx \tfrac{1}{2}\rho S(a_1\alpha_e + a_0)V_e^2, \tag{4.11}$$

$$L = \tfrac{1}{2}\rho S(b_1\alpha_e + b_0)V_e^2, \tag{4.12}$$

where S is the wing area, a_0, a_1, a_2 and b_0, b_1 are constant parameters.

4.4 Convex optimisation

This chapter considers how to robustly generate minimum power trajectories for the transition between powered lift and cruise flight modes, suggesting the following objective

$$J = \int_{t_0}^{t_f} P/\overline{P}\ \mathrm{d}t, \tag{4.13}$$

where $P = TV\cos\alpha$ is the drive power and $\overline{P} = \overline{TV}$. The optimisation problem consists of minimising (4.13) while satisfying dynamical constraints, input and state constraints (4.1)-(4.12). As such, this problem is a NLP and we thus consider below how to reformulate the problem as a sequence of convex programs. We introduce 4 key manipulations to do so: i) assuming that a path is known *a priori*, we introduce a change of differential operator to integrate the EOM over space, thus simplifying the structure of the problem; ii) to reduce the couplings between the optimisation variables, we combine both EOM to separate the optimisation of the velocity and torque from the other variables, allowing us to solve 2 smaller optimisation problems sequentially and accelerate computation; iii) we discretise the problem; iv) we approximate the nonlinear dynamics by a difference of convex functions and exploit the fact that convex functions can be bounded tightly by convex and linear bounds.

4.4.1 Change of differential operator

Assuming that a path $(X(s), Z(s))$ parameterised by the curvilinear abscissa s is known *a priori* (which is usually the case in a UAM context where flight corridors are prescribed) and applying the change of differential operator [130] $\frac{\mathrm{d}}{\mathrm{d}t} = V\frac{\mathrm{d}}{\mathrm{d}s}, \forall V \neq 0$, the dynamics in (4.1)-(4.3) becomes

$$\tfrac{1}{2}mE' = T\cos\alpha - \tfrac{1}{2}\rho S\left(a_1\alpha_e + a_0\right)\left(E + \tfrac{2T}{\rho A n}\right) - mg\sin\gamma^*, \tag{4.14}$$

$$mE\gamma^{*\prime} = T\sin\alpha + \tfrac{1}{2}\rho S\left(b_1\alpha_e + b_0\right)\left(E + \tfrac{2T}{\rho A n}\right) - mg\cos\gamma^*, \tag{4.15}$$

$$J_w(\tfrac{1}{2}E'i_w' + Ei_w'') = M,\ i_w(s_0) = i_0,\ i_w'(s_0)\sqrt{E(s_0)} = \Omega_0 \tag{4.16}$$

where $\frac{\mathrm{d}\cdot}{\mathrm{d}s} = \cdot'$ and $E = V^2$. The flight path angle $\gamma^* = \arctan\left(-\mathrm{d}Z/\mathrm{d}X\right)$ is known *a priori* from the path.

4.4.2 Problem separation

We next reduce the couplings between the states and inputs in the EOM (4.14)-(4.16) by eliminating the angle of attack from the formulation and separating the optimisation into two subproblems as follows. Let $\lambda = a_1/b_1$, then the combination (4.14) + λ(4.15) yields

$$\frac{1}{2}mE' + \underbrace{(\lambda m\gamma^{*\prime} + \frac{1}{2}\rho S(a_0 - \lambda b_0))}_{c(\gamma^{*\prime})} E + \underbrace{mg(\sin\gamma^* + \lambda\cos\gamma^*)}_{d(\gamma^*)} = \underbrace{T\cos\alpha + \lambda T\sin\alpha - S^\star(a_0 - \lambda b_0)T}_{\tau}, \quad (4.17)$$

where $S^\star = \frac{S}{An}$ and τ is a virtual input defined by

$$\tau = T\cos\alpha + \lambda T\sin\alpha - S^\star(a_0 - \lambda b_0)T. \quad (4.18)$$

The state and input constraints in (4.6)-(4.8) can be rewritten as

$$0 \le E \le \overline{V}^2, \quad \underline{a} \le E'/2 \le \overline{a}, \quad 0 \le \tau \le \overline{T}, \quad (4.19)$$

$$E(s_0) = V_0^2 \text{ and } E(s_f) = V_f^2. \quad (4.20)$$

In the thrust constraint in (4.19), τ was chosen as a proxy for T since $\lambda \ll 1$, and $S^\star(a_0 - \lambda b_0) \ll 1$, implying $\tau \approx T\cos\alpha$. This results in the constraint $\tau \le \overline{T}$ being a relaxed version of the original (we note that the original thrust constraint is inactive in practice – see Section 5). Likewise, the minimum power criterion in (4.13) can be approximated by a convex objective function under these conditions. By the change of differential operator we obtain

$$J = \int_{s_0}^{s_f} \tau/\overline{P}\ \mathrm{d}s. \quad (4.21)$$

Since γ and γ' are prescribed by the path, (4.17) is a linear equality constraint and the following convex optimisation problem can be constructed to minimise (4.21) subject to (4.17), (4.19) and (4.20) as follows

$$\mathcal{P}_1 : \min_{\tau, E, a} \qquad \int_{s_0}^{s_f} \tau/\overline{P}\,\mathrm{d}s,$$

$$\begin{aligned} \text{s.t.} \quad & \frac{1}{2}mE' + c(\gamma^{*\prime})E + d(\gamma^*) = \tau, \\ & 0 \le \tau \le \overline{T},\ \underline{a} \le \frac{1}{2}E' \le \overline{a}, \\ & 0 \le E \le \overline{V}^2,\ E(s_0) = V_0^2,\ E(s_f) = V_f^2. \end{aligned}$$

Solving $\mathcal{P}_1$ yields the optimal velocity profile along the path and provides a proxy for the optimal thrust. However, a tiltwing angle profile that meets the dynamical constraints and follows the desired path with $\gamma \approx \gamma^*$ must also be computed. To achieve this we use the solution of $\mathcal{P}_1$ to define a new optimisation problem with variables γ, α, i_w, and M satisfying the constraints (4.5), (4.16) and, using (4.18) to eliminate the thrust in (4.15),

$$mE\gamma' = \underbrace{\tau \sin\alpha + \frac{1}{2}\rho S\left[b_1 \arcsin\left(\frac{\sqrt{E}\sin\alpha}{\sqrt{E + \frac{2\tau}{\rho A n}}}\right) + b_0\right]\left(E + \frac{2\tau}{\rho A n}\right)}_{f(\alpha, E, \tau)} - mg\cos\gamma, \quad (4.22)$$

in which the objective is to minimise the cost function

$$J_\gamma = \int_{s_0}^{s_f} \frac{(\gamma - \gamma^*)^2}{\sqrt{E}}\, \mathrm{d}s. \quad (4.23)$$

Note that only the two EOM (4.15) and (4.16) are needed to construct this new problem since the linear combination (4.14) + λ (4.15) is enforced with τ and E prescribed from problem $\mathcal{P}_1$. We thus state the following optimisation

$$\begin{aligned} \mathcal{P}_2 : \min_{\alpha, \gamma, i_w, M} \quad & \int_{s_0}^{s_f} \frac{(\gamma - \gamma^*)^2}{\sqrt{E}}\, \mathrm{d}s \\ \text{s.t.} \quad & mE\gamma' = f(\alpha, E, \tau) - mg\cos\gamma, \\ & J_w(\tfrac{1}{2}E' i_w' + i_w'' E) = M,\ i_w(s_0) = i_0, \\ & i_w'(s_0)\sqrt{E(s_0)} = \Omega_0, \\ & i_w = \alpha + \gamma, \\ & \underline{M} \le M \le \overline{M},\ \underline{\alpha} \le \alpha \le \overline{\alpha} \\ & \underline{\gamma} \le \gamma \le \overline{\gamma},\ \underline{i_w} \le i_w \le \overline{i_w}, \end{aligned}$$

and reconstruct the input T and state V *a posteriori* using (4.18) and $V = \sqrt{E}$. Given the solution of both problems as functions of the independent variable s, the final step is to map the solution to time domain by reversing the change of differential operator and integrating $t(\xi) = \int_{s_0}^{\xi} \frac{\mathrm{d}s}{V(s)}$.

4.4.3 Discretisation

To obtain computationally tractable problems, we consider $N+1$ discretisation points $\{s_0, s_1, \ldots, s_N\}$ of the path, with spacing $\delta_k = s_{k+1} - s_k$, $k = 0, \ldots, N-1$.

Assuming a path $s_k \to (X_k, Z_k)$, the prescribed flight path angle and rate are discretised as follows

$$\gamma_k^* = \arctan\Big(-\frac{Z_{k+1} - Z_k}{X_{k+1} - X_k}\Big), \quad k \in \{0, \ldots, N-1\}, \tag{4.24}$$

$$\gamma_k^{*\prime} = \begin{cases} (\gamma_{k+1}^* - \gamma_k^*)/\delta_k, & k \in \{0, \ldots, N-2\}, \\ \gamma_{N-2}^{*\prime}, & k = N-1. \end{cases} \tag{4.25}$$

The resulting discretised versions of $\mathcal{P}_1$ and $\mathcal{P}_2$ are

$$\begin{aligned}
\mathcal{P}_1^\dagger : \min_{\tau,\, E,\, a} \quad & \sum_{k=0}^{N-1} \tau_k / \overline{P}\, \delta_k, \\
\text{s.t.} \quad & E_{k+1} = E_k + \frac{2\delta_k}{m}(\tau_k - c(\gamma_k^{*\prime})E_k - d(\gamma_k^*)), \\
& 0 \le \tau_k \le \overline{T}, \quad \underline{a} \le \frac{E_{k+1} - E_k}{2\delta_k} \le \overline{a}, \\
& 0 \le E_k \le \overline{V}^2, \quad E_0 = V_0^2, \quad E_N = V_f^2, \\
\mathcal{P}_2^\dagger : \min_{\substack{\alpha,\gamma,\\ i_w,\,\zeta,\,M}} \quad & \sum_{k=0}^{N-1} \frac{(\gamma_k - \gamma_k^*)^2}{\sqrt{E_k}} \delta_k \\
\text{s.t.} \quad & \gamma_{k+1} = \gamma_k + \frac{\delta_k}{mE}(f_k(\alpha_k, E_k, \tau_k) - f_{\gamma_k}(\gamma_k)), \\
& i_{w,k} = \alpha_k + \gamma_k, \\
& i_{w,k+1} = i_{w,k} + \zeta_k \delta_k, \quad i_{w,0} = i_0, \\
& \zeta_{k+1} = \zeta_k\Big(1 - \frac{E_{k+1} - E_k}{2E_k}\Big) + \frac{M_k \delta_k}{J_w E_k}, \\
& \zeta_0 \sqrt{E_0} = \Omega_0, \\
& \underline{M} \le M_k \le \overline{M}, \quad \underline{\alpha} \le \alpha_k \le \overline{\alpha}, \\
& \underline{\gamma} \le \gamma_k \le \overline{\gamma}, \quad \underline{i_w} \le i_{w,k} \le \overline{i_w}.
\end{aligned}$$

where $f_{\gamma_k} = -mg\cos\gamma_k$. The input and state variables are reconstructed using

$$T_k = \frac{\tau_k}{\cos\alpha_k + \lambda \sin\alpha_k - S^\star(a_0 - \lambda b_0)}, \quad V_k = \sqrt{E_k}, \tag{4.26}$$

and the time t_k associated with each discretisation point is computed, allowing solutions to be expressed as time series

$$t_k = \sum_{j=0}^{k-1} \frac{\delta_j}{V_j}. \tag{4.27}$$

We now have a pair of finite dimensional problems $\mathcal{P}_1^\dagger$ and $\mathcal{P}_2^\dagger$, but the latter is still nonconvex due to the nonlinear equality constraint involving functions f_k and f_{γ_k}.

One approach would be to crudely linearise that constraint, but this would yield to solutions for the trajectory that are not feasible for the original nonlinear problem or to instability as identified in [129].

Instead, we propose a new approach based on differences of convex (DC) function decomposition. The idea is that, if the dynamical constraint can be reformulated in DC form, the trajectories of the system could be bounded tightly by a set of convex inequalities as discussed in Section 4.4.5. This is due to the mathematical properties of convex functions and their (necessarily convex) linearisation errors (see Appendix A). Although the proposed method still involves linearisation, the effect of the linearisation error is limited since it can be bounded tightly and treated as a bounded disturbance in a robust optimisation framework, exploiting the curvature properties of the function. Moreover, the approach we propose results in a solution that is feasible for the original problem (see Chapter 5).

On a restricted domain $\gamma_k \in [-\pi/2; \pi/2]$, f_{γ_k} is a convex function of γ_k, making it possible to derive tight convex bounds on f_{γ_k}. However, this is not the case with f_k and we now introduce a method to obtain a DC decomposition of this function.

4.4.4 DC decomposition

We seek a decomposition of f_k as a difference of convex (DC) functions: $f_k = g_k - h_k$, where g_k, h_k are convex. A DC decomposition always exists if $f_k \in \mathcal{C}^2$ [127].

Note that since (E_k, τ_k) are obtained from problem $\mathcal{P}_1^\dagger$, the function $f_k(\alpha_k, E_k, \tau_k)$ is single-valued (in α_k) which considerably simplifies the task of finding a DC decomposition, and motivates the above approach of separating the initial problem in two subproblems with fewer couplings between the variables. However, f_k is also time varying through its dependence on parameters (E_k, τ_k) generated online. This requires us to find a DC split for every instance of (E_k, τ_k), $\forall k \in [0, \ldots, N]$ which can be intractable if the horizon is large or the sampling interval is small.

Instead, we adopt the more pragmatic approach of i) precomputing offline the DC decompositions on a downsampled grid of values (E_i, τ_j), $\forall (i,j) \in [0,\dots,N_s] \times [0,\dots,M_s]$ where $N_s, M_s \ll N+1$ and ii) interpolating the obtained decompositions online using a look-up table.

Precomputation of the DC decomposition

Inspired by [131], we develop a computationally tractable method for the DC decomposition of a function $f_k(\alpha)$ based on approximating[2] the function by a polynomial of degree $2n$:

$$f_k(\alpha) \approx p_{k,2n}\alpha^{2n} + ... + p_{k,1}\alpha + p_{k,0} = y^\top F_k y, \tag{4.28}$$

where $y = [1 \quad \alpha \quad ... \quad \alpha^n]^\top \in \mathbb{R}^{n+1}$ is a vector of monomials of increasing order and $F_k = F_k^\top \in R^{n+1\times n+1}$ is the Gram matrix of the polynomial defined by $\{F_k\}_{ij} = p_{k,i+j+1}/\lambda(i,j)$, $\forall i,j \in [0,1,...,n]$ where $\lambda(i,j) = i+j+1$ if $i+j \leq n$ and $\lambda(i,j) = 2n+1-(i+j)$. Given N_s samples $f_{k,s}\, \forall s \in [1,...,N_s]$ of the function f_k, the polynomial approximation can be obtained by solving a least square problem to find the coefficients that best fit the samples.

We now seek the symmetric matrices G_k, H_k such that

$$y^\top F_k y = y^\top G_k y - y^\top H_k y,$$

where $g_k \approx y^\top G_k y$ and $h_k \approx y^\top H_k y$ are convex polynomials in α. Such conditions can be satisfied if the Hessians $d^2 g_k/d\alpha^2 = y^\top H_{g_k} y$ and $d^2 h_k/d\alpha^2 = y^\top H_{h_k} y$ satisfy the following linear matrix inequalities (LMI),

$$H_{g_k} \equiv {D^\top}^2 G_k + G_k D^2 + 2D^\top G_k D \succeq 0,$$
$$H_{h_k} \equiv {D^\top}^2 H_k + H_k D^2 + 2D^\top H_k D \succeq 0,$$

where D is a matrix of coefficients such that $dy/d\alpha = Dy$. Finding the DC decomposition thus reduces to solving the following semidefinite program (SDP)

$$\mathcal{SDP} : \min_{H_{g_k}} \qquad \operatorname{tr} H_{g_k}$$

[2]Note that any continuous function can be approximated arbitrarily closely by a polynomial, in agreement with Stone-Weierstrass approximation theorem.

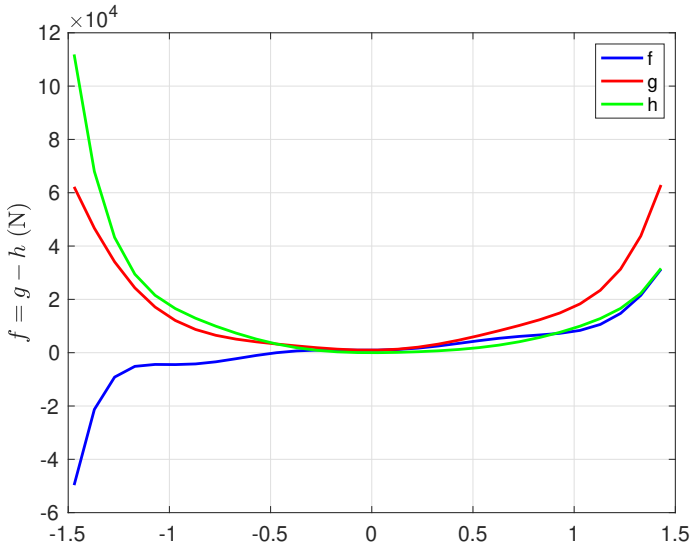


Figure 4.2: Example DC decomposition of polynomial approximation of $f_k(\alpha)$.

$$\begin{aligned} \text{s.t.} \qquad & H_{g_k} \succeq 0, \\ & H_{g_k} - ({D^\top}^2 F_k + F_k D^2 + 2D^\top F_k D) \succeq 0, \end{aligned}$$

and computing $H_{h_k} = H_{g_k} - {D^\top}^2 F_k + F_k D^2 + D^\top F_k D$, followed by the double integration $d^2 g_k / d\alpha^2 = y^\top H_{g_k} y$ and $d^2 h_k / d\alpha^2 = y^\top H_{h_k} y$ to recover g_k and h_k. This operation is repeated at each point (E_i, τ_j) of the grid to assemble a look-up table of polynomial coefficients. Note that the objective was chosen so as to regularise the solutions for g_k, h_k by minimising a proxy for their average curvature.

A typical DC decomposition of the polynomial approximation for a given (E_i, τ_j) is shown in Figure 4.2.

Coefficient interpolation

A bilinear interpolation of the coefficients is performed online to obtain the DC decomposition for each (E_k, τ_k) such that $E_i \leq E_k < E_{i+1}$ and $\tau_j \leq \tau_k < \tau_{j+1}$ as follows ($\forall l \in \{1, 2\}$, $f_1 \equiv g$, $f_2 \equiv h$):

$$\begin{aligned} f_{l,k} = w_{i,j} f_l(\alpha, E_i, \tau_j) + w_{i+1,j} f_l(\alpha, E_{i+1}, \tau_j) \\ + w_{i,j+1} f_l(\alpha, E_i, \tau_{j+1}) + w_{i+1,j+1} f_l(\alpha, E_{i+1}, \tau_{j+1}), \end{aligned} \tag{4.29}$$

where

$$w_{i,j} = (E_{i+1} - E_k)(\tau_{j+1} - \tau_k)/(\Delta E \Delta \tau),$$

$$w_{i+1,j} = (E_k - E_i)(\tau_{j+1} - \tau_k)/(\Delta E \Delta\tau),$$
$$w_{i,j+1} = (E_{i+1} - E_k)(\tau_k - \tau_j)/(\Delta E \Delta\tau),$$
$$w_{i+1,j+1} = (E_k - E_i)(\tau_k - \tau_j)/(\Delta E \Delta\tau),$$

with $\Delta E = E_{i+1} - E_i$ and $\Delta\tau = \tau_{j+1} - \tau_j$.

This operation preserves convexity since the interpolated polynomial coefficients are a weighted sum of the coefficients in the lookup table.

4.4.5 Convex relaxation

Consider again the nonlinear dynamics in problem $\mathcal{P}_2^\dagger$, using the DC decomposition of f_k computed in the previous section and eliminating the angle of attack via $\alpha_k = i_{w,k} - \gamma_k$ to reduce the number of states, we obtain

$$\gamma_{k+1} = \gamma_k + \frac{\delta_k}{mE}(g_k(i_{w,k} - \gamma_k, E_k, \tau_k) - h_k(i_{w,k} - \gamma_k, E_k, \tau_k) - mg\cos\gamma_k). \quad (4.30)$$

All nonlinearities in (4.30) involve convex and concave functions of the states $i_{w,k}$ and γ_k, with dynamics

$$i_{w,k+1} = i_{w,k} + \zeta_k\delta_k, \quad (4.31)$$
$$\zeta_{k+1} = \zeta_k\Big(1 - \frac{E_{k+1} - E_k}{2E_k}\Big) + \frac{M_k\delta_k}{J_w E_k}. \quad (4.32)$$

In what follows we will exploit the convexity properties of the functions g_k, h_k, $f_{\gamma_k} = -mg\cos\gamma_k$ in (4.30) to approximate the dynamics by a set of convex inequalities with tight bounds on the state trajectories. To do so, we linearise the dynamics successively around feasible guessed trajectories and treat the linearisation error as a bounded disturbance [83]. We use the fact that the linearisation error of a convex (resp. concave) function is also convex[3] (resp. concave) and can thus be bounded tightly since its maximum (resp. minimum) occurs at the boundary of the set on which the function is constrained[4]. This allows us to construct a robust optimisation and to obtain solutions that are robust to the linearisation error.

[3]Appendix A, Theorem 14.
[4]Ibid., Theorem 16.

We start by assuming the existence of a set of feasible trajectories $i_{w,k}$ and γ_k for (4.30)-(4.32) and consider the perturbed dynamics

$$\gamma_{k+1} = \gamma_k + \frac{\delta_k}{mE}(g_k^\circ + \nabla g_k^\circ(i_{w,k} - \gamma_k - (i_{w,k}^\circ - \gamma_k^\circ)) + w_1$$
$$-h_k^\circ - \nabla h_k^\circ(i_{w,k} - \gamma_k - (i_{w,k}^\circ - \gamma_k^\circ)) - w_2 - mg\cos\gamma_k^\circ + mg\sin\gamma_k^\circ(\gamma_k - \gamma_k^\circ) + w_3). \tag{4.33}$$

where $g_k^\circ = g_k(i_{w,k}^\circ - \gamma_k^\circ)$, $h_k^\circ = h_k(i_{w,k}^\circ - \gamma_k^\circ)$ are the functions g_k, h_k evaluated along the guessed trajectory, $\nabla g_k^\circ = dg_k/d\alpha_k(i_{w,k}^\circ - \gamma_k^\circ)$, $\nabla h_k^\circ = dh_k/d\alpha_k(i_{w,k}^\circ - \gamma_k^\circ)$ are the first order derivatives of g_k, h_k evaluated along the guessed trajectory, and $w_1(i_w - \gamma, i_{w,k}^\circ - \gamma_k^\circ)$, $w_2(i_w - \gamma, i_{w,k}^\circ - \gamma_k^\circ)$, $w_3(\gamma, \gamma_k^\circ)$ are the convex linearisation errors of $g_k, h_k, f_{\gamma_k} = -mg\cos\gamma_k$ respectively. Since these linearisation errors are convex, they take their maximum on the boundary of the set over which the functions are constrained. Moreover, by definition, their minimum on this set is zero (Jacobian linearisation). We thus infer the following relationships $\forall i = \{1, 2\}$ and noting $f_1 \equiv g, f_2 \equiv h$

$$\min_{\substack{\gamma\in[\underline{\gamma}_k,\overline{\gamma}_k]\\ i_w\in[\underline{i}_{w,k},\overline{i}_{w,k}]}} w_i(i_w - \gamma, i_w^\circ - \gamma^\circ) = 0, \tag{4.34}$$

$$\min_{\gamma\in[\underline{\gamma}_k,\overline{\gamma}_k]} w_3(\gamma, \gamma^\circ) = 0, \tag{4.35}$$

$$\max_{\substack{\gamma\in[\underline{\gamma}_k,\overline{\gamma}_k]\\ i_w\in[\underline{i}_{w,k},\overline{i}_{w,k}]}} w_i(i_w - \gamma, i_w^\circ - \gamma^\circ) = \max\{f_{i,k} - f_{i,k}^\circ - \nabla f_{i,k}^\circ(\overline{i}_{w,k} - \underline{\gamma}_k - (i_{w,k}^\circ - \gamma_k^\circ));$$
$$f_{i,k} - f_{i,k}^\circ - \nabla f_{i,k}^\circ(\underline{i}_{w,k} - \overline{\gamma}_k - (i_{w,k}^\circ - \gamma_k^\circ))\}, \tag{4.36}$$

$$\max_{\gamma\in[\underline{\gamma}_k,\overline{\gamma}_k]} w_3(\gamma, \gamma^\circ) = \max\{-mg\cos\underline{\gamma}_k + mg\cos\gamma_k^\circ - mg\sin\gamma_k^\circ(\underline{\gamma}_k - \gamma_k^\circ);$$
$$-mg\cos\overline{\gamma}_k + mg\cos\gamma_k^\circ - mg\sin\gamma_k^\circ(\overline{\gamma}_k - \gamma_k^\circ)\} \tag{4.37}$$

where we assumed that the state trajectories γ_k and $i_{w,k}$ lie within "tubes" whose cross-sections are parameterised by means of elementwise bounds $\gamma_k \in [\underline{\gamma}_k, \overline{\gamma}_k]$ and $i_{w,k} \in [\underline{i}_{w,k}, \overline{i}_{w,k}]$, $\forall k$, which are considered to be optimisation variables. Given these bounds on the states at a given time instant and by virtue of equations (4.34)-(4.37), the bounds on the states at the next time instant satisfy the following

convex inequalities

$$\overline{\gamma}_{k+1} \geq \max_{\substack{\gamma\in\{\underline{\gamma}_k;\overline{\gamma}_k\}\\ i_w\in\{\underline{i}_{w,k};\overline{i}_{w,k}\}}} \Big\{\gamma+\tfrac{\delta_k}{mE}(g_k(i_w-\gamma)-h_k(i_w^\circ-\gamma^\circ)-\nabla h_k^\circ(i_w-\gamma-(i_{w,k}^\circ-\gamma_k^\circ)) - mg\cos\gamma)\Big\}, \tag{4.38}$$

$$\underline{\gamma}_{k+1} \leq \min_{\substack{\gamma\in\{\underline{\gamma}_k,\overline{\gamma}_k\}\\ i_w\in\{\underline{i}_{w,k},\overline{i}_{w,k}\}}} \Big\{\gamma+\tfrac{\delta_k}{mE}(g_k(i_w^\circ-\gamma^\circ)+\nabla g_k^\circ(i_w-\gamma-(i_{w,k}^\circ-\gamma_k^\circ))-h_k(i_w-\gamma) -mg\cos\gamma^\circ+mg\sin\gamma^\circ(i_w-\gamma))\Big\}, \tag{4.39}$$

$$\overline{i}_{w,k+1}\geq\overline{i}_{w,k}+\zeta_k\delta_k,\quad \underline{i}_{w,k+1}\leq\underline{i}_{w,k}+\zeta_k\delta_k. \tag{4.40}$$

These conditions involve only minimisations of linear functions and maximisations of convex functions, which can be solved simply by evaluating the functions at the boundary of the set[5]. Hence each inequality reduces to $2^2=4$ convex inequalities (obtained by replacing γ with $\{\underline{\gamma}_k,\overline{\gamma}_k\}$ and i_w with $\{\underline{i}_{w,k},\overline{i}_{w,k}\}$). Moreover, this number can be reduced, avoiding the curse of dimensionality, since the coefficients of the linear functions appearing in each maximisation and minimisation are known. Finally, the required computation can be further reduced by introducing a low-order approximation of the polynomials in (4.38)-(4.40). This was obtained by computing, before including the constraints in the optimisation, a series of quadratic polynomials to each g_k, h_k, $\forall k$ that are a best fit around $i_{w,k}^\circ-\gamma_k^\circ$.

The tube defined by inequalities (4.38)-(4.40) can be used to replace $\mathcal{P}_2^\dagger$ by a sequence of convex programs. Given the solution of $\mathcal{P}_1^\dagger$ and given a set of feasible (suboptimal) trajectories $i_{w,k}^\circ$, γ_k° satisfying (4.30)-(4.32), the following convex problem is solved sequentially

$$\begin{aligned}\mathcal{P}_2^\ddagger:\ \min_{\substack{\overline{\gamma},\underline{\gamma},\overline{i}_w,\\ \underline{i}_w,\overline{\zeta},M,\theta}}\quad & \sum_{k=0}^{N-1}\frac{\theta_k^2}{\sqrt{E_k}}\delta_k\\ \text{s.t.}\quad & \theta_k\geq|\overline{\gamma}_k-\gamma_k^*|,\quad \theta_k\geq|\underline{\gamma}_k-\gamma_k^*|,\\ & \overline{\gamma}_{k+1}\geq\max_{\substack{\gamma\in\{\underline{\gamma}_k;\overline{\gamma}_k\}\\ i_w\in\{\underline{i}_{w,k};\overline{i}_{w,k}\}}}\Big\{\gamma+\frac{\delta_k}{mE}(g_k(i_w-\gamma)\end{aligned}$$

[5]Ibid., Lemma 15

$$
\begin{aligned}
&- h_k(i_w^\circ - \gamma^\circ) - \nabla h_k^\circ(i_w - \gamma - (i_{w,k}^\circ - \gamma_k^\circ)) - mg\cos\gamma)\Big\},\\
\underline{\gamma}_{k+1} \leq &\min_{\substack{\gamma\in\{\underline{\gamma}_k,\overline{\gamma}_k\}\\ i_w\in\{\underline{i}_{w,k},\bar{i}_{w,k}\}}} \Big\{\gamma + \frac{\delta_k}{mE}(g_k(i_w^\circ - \gamma^\circ)\\
&+ \nabla g_k^\circ(i_w - \gamma - (i_{w,k}^\circ - \gamma_k^\circ)) - h_k(i_w - \gamma)\\
&- mg\cos\gamma^\circ + mg\sin\gamma^\circ(i_w - \gamma))\Big\},\\
&\bar{i}_{w,k+1} \geq \bar{i}_{w,k} + \zeta_k\delta_k, \quad \bar{i}_{w,0} = i_0,\\
&\underline{i}_{w,k+1} \leq \underline{i}_{w,k} + \zeta_k\delta_k, \quad \underline{i}_{w,0} = i_0,\\
&\zeta_{k+1} = \zeta_k\Big(1 - \frac{E_{k+1} - E_k}{2E_k}\Big) + \frac{M_k\delta_k}{J_w E_k},\\
&\zeta_0\sqrt{E_0} = \Omega_0,\\
&\underline{M} \leq M_k \leq \overline{M}, \quad \underline{i_w} \leq \underline{i}_{w,k}, \quad \bar{i}_{w,k} \leq \overline{i_w},\\
&\underline{\alpha} \leq \underline{i}_{w,k} - \overline{\gamma}_k, \quad \bar{i}_{w,k} - \underline{\gamma}_k \leq \overline{\alpha},\\
&\underline{\gamma} \leq \underline{\gamma}_k, \quad \overline{\gamma}_k \leq \overline{\gamma}, \quad \overline{\gamma}_0 = \gamma_0, \quad \underline{\gamma}_0 = \gamma_0.
\end{aligned}
$$

After each iteration of this problem, the guessed trajectories are updated by applying M_k to (4.31)-(4.32), and updating

$$E_{k+1} = E_k + \frac{2\delta_k}{m}(T_k\cos\alpha_k - D_k - mg\sin\gamma_k), \tag{4.41}$$

$$\gamma_{k+1} = \gamma_k + \frac{\delta_k}{mE}(f_k(\alpha_k, E_k, \tau_k) - f_{\gamma_k}(\gamma_k)), \tag{4.42}$$

where $D_k = \frac{1}{2}\rho S(a_2\alpha_e^2 + a_1\alpha_e + a_0)V_e^2$ and T_k is obtained using equation (4.26). The process is repeated until $|\overline{\gamma}_k - \underline{\gamma}_k|$ and $|\bar{i}_{w,k} - \underline{i}_{w,k}|$ have converged. Once $\mathcal{P}_1^\dagger$ and $\mathcal{P}_2^\ddagger$ have been solved, we check whether $|\gamma_k^* - \gamma_k| \leq \epsilon\ \forall k \in \{0, \ldots, N\}$, where ϵ is a specified tolerance. If this condition is not met ($\mathcal{P}_2^\ddagger$ may admit solutions that allow γ_k to differ from the assumed flight path angle γ_k^*), the problem is reinitialized with the updated flight path angle and rate $\gamma_k^* \leftarrow \gamma_k$, $\gamma_k^{*\prime} \leftarrow \gamma_k'$ and $\mathcal{P}_1^\dagger$ and $\mathcal{P}_2^\ddagger$ are solved again. When the solution tolerance is met (or the maximum number of iterations is exceeded) the problem is considered solved and the input and state variables are reconstructed using the equations in (4.26) and the time t_k associated with each discretisation point is computed with (4.27), allowing solutions to be expressed as time series. The procedure is summarised in Algorithm 1.

Remarks on $\mathcal{P}_2^\ddagger$: i) the angle of attack has been eliminated from the formulation; ii) to ensure convexity, it is important that $\overline{\gamma} \leq \pi/2$ and $\underline{\gamma} \geq -\pi/2$; iii) order reduction was performed on polynomials g_k, h_k, i.e. quadratic polynomials were fitted to g_k, h_k around $i_{w,k}^\circ - \gamma_k^\circ$ for all k by solving a least squares problem before running the optimisation; iv) to improve numerical stability, M_k can be replaced by $M_k + K_{i,k}(i_{w,k} - i_{w,k}^\circ) + K_{\zeta,k}(\zeta_k - \zeta_k^\circ)$ with $i_{w,k} \in \{\underline{i}_{w,k}; \overline{i}_{w,k}\}$ where $K_{i,k}$ and $K_{\zeta,k}$ are obtained, e.g. by solving a LQR problem for the time varying linear system in (4.31)-(4.32); v) the slack variable θ_k was introduced to enforce the objective $\sum_k \max_{\gamma_k \in \{\underline{\gamma}_k; \overline{\gamma}_k\}} (\gamma_k - \gamma^*)^2/\sqrt{E_k}$. Note also that the maximisation operation reduces to evaluating the objective at the vertices of the polytopic set by convexity of the objective, see Appendix A, Lemma 15.

4.5 Results

We consider a case study based on the Airbus A^3 Vahana. The aircraft parameters are reported in Table 4.1. We run Algorithm 1 using the convex programming software package CVX [123] with the solver Mosek [132] to compute the optimal trajectory for the transition manoeuvre with boundary conditions given as follows: initial and final velocity $\{V_0; V_f\} = \{0.5; 40\}$ m/s, tiltwing angle $i_0 = 75$ deg, tiltwing angle rate $\Omega_0 = 0$ deg/s, flight path angle $\gamma_0 = 0$ deg. We limit the number of iterations of problem $\mathcal{P}_1^\dagger$ to 1 and of $\mathcal{P}_2^\ddagger$ to 3.

The time to completion as a function of problem size (number of discretisation points) is shown in Figure 4.3, implying excellent scalability. In particular, for $N < 1500$, the time to completion is of the order of seconds. A real-time solution is thus possible using modest computational resources. This is in stark contrast to the complexity of generic NLP approaches (e.g. in [62], computation times of the order of minutes are quoted to solve a similar VTOL trajectory optimisation problem using the OpenMDAO framework and SNOPT). Moreover, as shown in Chapter 3, the time complexity for this type of problem can be reduced significantly with first-order solvers, paving the way for fast real-time implementations.

Algorithm 1: Convex trajectory optimisation

Compute the DC decomposition of f_k on grid of points (E_i, τ_j):
approximate f_k by a polynomial (4.28) and solve $\mathcal{SDP}$ at each point to obtain g_k, h_k and build a look-up table of polynomial coefficients.
Compute γ^*, $\gamma^{*\prime}$ using (4.24), (4.25) and initialise: $\gamma \leftarrow \gamma^*$, $\gamma' \leftarrow \gamma^{*\prime}$, $\gamma^* \leftarrow \infty$, $j \leftarrow 0$
while $\max_{k\in\{0,\dots,N\}} |\gamma_k^* - \gamma_k| > \epsilon$ & $j < MaxIters$ **do**
    $\gamma^* \leftarrow \gamma$, $\gamma^{*\prime} \leftarrow \gamma'$
    Solve problem $\mathcal{P}_1^\dagger$.
    Compute the gains $K_{\zeta,k}$ and $K_{i,k}$.
    Compute feasible trajectories $i_{w,k}^\circ$ and γ_k°.
    $i \leftarrow 0$
    while $\max_{k\in\{0,\dots,N\}} |\overline{\gamma}_k - \underline{\gamma}_k| > \epsilon$ & $\max_{k\in\{0,\dots,N\}} |\bar{i}_{w,k} - \underline{i}_{w,k}| > \epsilon$ & $i < MaxIters$ **do**
        for $k \leftarrow 0$ **to** N **do**
            Interpolate g_k, h_k from the look-up table.
            Fit a quadratic model to the interpolated g_k, h_k that is a best fit at $i_{w,k}^\circ - \gamma_k^\circ$.
        end
        Solve problem $\mathcal{P}_2^\ddagger$.
        Update guess trajectories with (4.31) - (4.32) and (4.41) - (4.42).
        $i \leftarrow i + 1$
    end
    $j \leftarrow j + 1$
end
for $k \leftarrow 0$ **to** N **do**
    $T_k \leftarrow \frac{\tau_k}{\cos\alpha_k + \lambda \sin\alpha_k - S^\star(a_0 - \lambda b_0)}$
    $V_k \leftarrow \sqrt{E_k}$, $t_k \leftarrow \sum_{j=0}^{k-1} \frac{\delta_j}{V_j}$
end

The simulation results are illustrated in Figure 4.4. The manoeuvre is a (near) constant altitude forward transition that requires a zero flight path angle throughout. As the aircraft transitions from powered lift to cruise, the velocity magnitude increases (a) and the thrust decreases (b), illustrating the change in lift generation from propellers to wing. The tiltwing angle drops quickly at the beginning (c), resulting in an increase in the angle of attack (d). The slight discrepancy in the flight path angle curves in (c) illustrates that problem $\mathcal{P}_2^\ddagger$ needs not necessarily generate a flight path angle profile corresponding to the exact desired path if the latter is

not feasible. Note from graph (d) that the effective angle of attack α_e stays within reasonable bounds, indicating that the wing is not stalled and that our assumption $\alpha_e << 1$ is valid. The angle of attack α however is not constrained to small values and we can thus achieve transitions at near constant altitude, see Figure 4.5.

Convergence of problem $\mathcal{P}_2^\ddagger$ after 3 iterations is shown in Figure 4.6. The tube bounds and objective converge quickly toward infinitesimal values after just a few iterations.

Parameter	**Symbol**	**Value**	**Units**
Mass	m	752.2	kg
Gravity acceleration	g	9.81	$\mathrm{m\,s^{-2}}$
Wing area	S	8.93	$\mathrm{m^2}$
Disk area	A	2.83	$\mathrm{m^2}$
Wing inertia	J_w	1100	$\mathrm{kg\,m^2}$
Density of air	ρ	1.225	$\mathrm{kg\,m^{-3}}$
Lift coefficients	b_0, b_1	$0.43, 0.11$	$-$, $\mathrm{deg^{-1}}$
Drag coefficients	a_0	0.02	$-$
	a_1	0.004	$\mathrm{deg^{-1}}$
	a_2	5.6e−4	$\mathrm{deg^{-2}}$
Maximum thrust	$\overline{T}$	8855	N
Angle of attack range	$[\underline{\alpha}, \overline{\alpha}]$	$[-90, 90]$	deg
Flight path angle range	$[\underline{\gamma}, \overline{\gamma}]$	$[-90, 90]$	deg
Tiltwing angle range	$[\underline{i}_w, \overline{i}_w]$	$[0, 100]$	deg
Acceleration range	$[\underline{a}, \overline{a}]$	$[-0.3g, 0.3g]$	$\mathrm{m\,s^{-2}}$
Velocity range	$[\underline{V}, \overline{V}]$	$[0, 40]$	m/s
Momentum range	$[\underline{M}, \overline{M}]$	$[-50; 50]$	N m
Number of propellers	n	4	$-$
Discretisation points	N	1000	$-$
Time step	δ	0.5	s
Degree of polynomial f	$2n$	26	$-$

Table 4.1: Model parameters derived from A^3 Vahana

4.6 Conclusions

This chapter addresses the trajectory optimisation problem for the transition of a tiltwing VTOL aircraft. A new robust optimisation paradigm was presented that leverages DC decomposition of the dynamics. The approach is based on

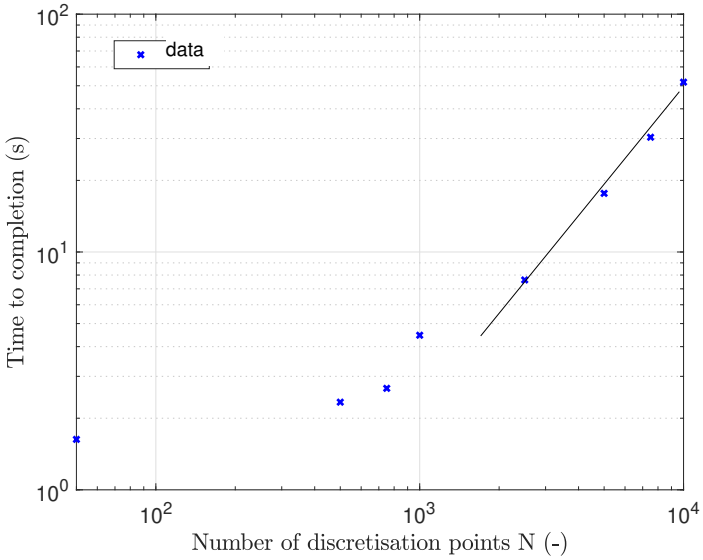


Figure 4.3: Time to completion as a function of problem size

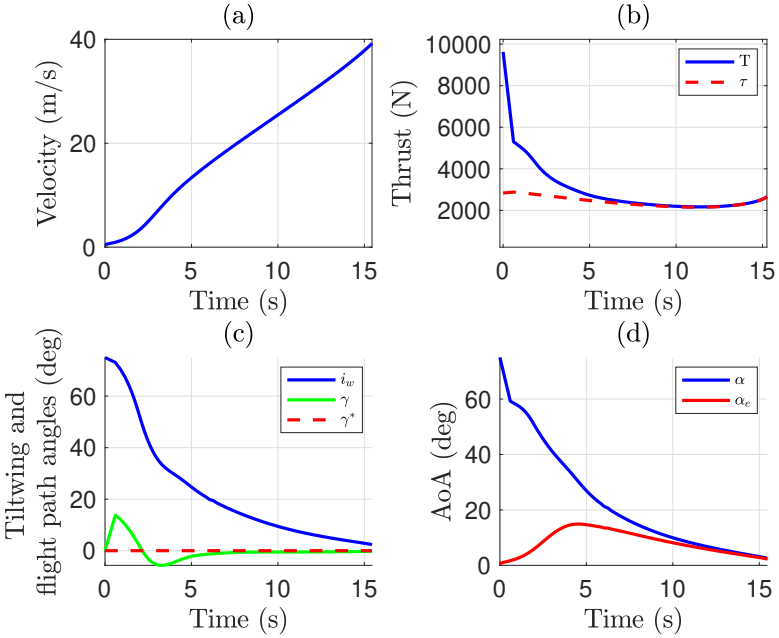


Figure 4.4: Forward transition.

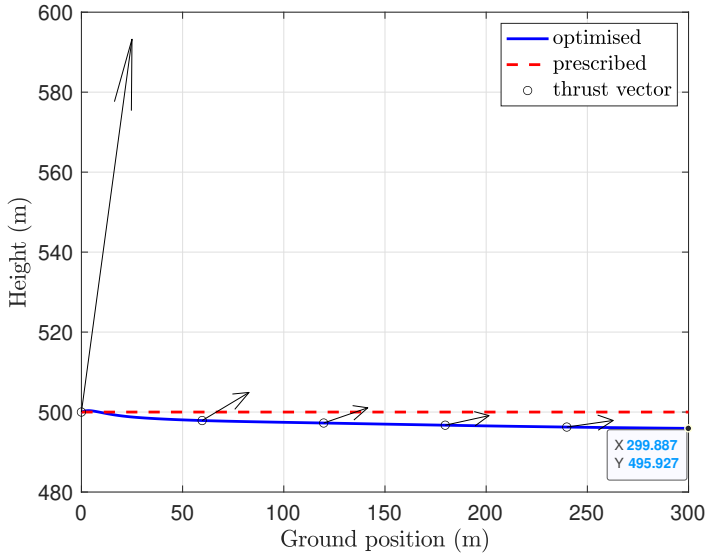


Figure 4.5: Altitude variation and thrust vector field during the forward transition.

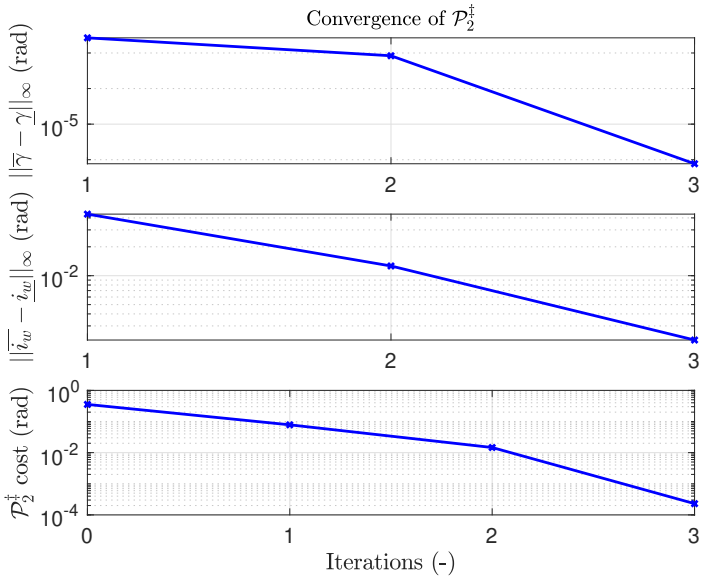


Figure 4.6: Convergence of tube bounds and objective for $\mathcal{P}_2^\ddagger$.

successive linearisation of the dynamics around feasible trajectories and treating the linearisation error as a bounded disturbance. The DC form of the dynamics allows to enforce tight bounds on the disturbance via a set of convex inequalities that form the basis of a computationally tractable robust optimisation. The algorithm can compute safe trajectories that are robust to model uncertainty for abrupt transitions at near constant altitude. Another contribution of the present work is a computationally tractable robust optimisation paradigm for dynamic systems that are not necessarily convex, by means of a DC decomposition of the nonlinear dynamics. In fact, the method could potentially be applied to any nonlinear system with twice continuously differentiable dynamics, so that our approach could be used to solve robust optimisation problems arising in numerous areas of research.

Limitations of the present approach are i) to obtain a computationally tractable formulation, quadratic approximations of the DC polynomials are required; ii) the computation time, although representing a considerable improvement compared to solving a NLP, is still too high to leverage the optimisation in real time for large problem dimensions.

Future work will alleviate these problems by i) considering other types of basis functions for the nonlinear dynamics approximation, e.g. radial basis functions or neural networks that have better scalability than a monomial basis (we will discuss this briefly in Chapter 5, Section 5.7.2); ii) investigating the use of first order solvers such as the alternating direction method of multipliers (ADMM) to accelerate computations as in Chapter 3; iii) parameterising the uncertainty sets with simplexes that scale linearly with the number of states to reduce the number of constraints.

Finally we note that the method presented could be leveraged in a multi-stage optimal control such as tube-based MPC (TMPC) to generate a safe and computationally tractable control law, e.g. for the transition of tiltwing VTOL aircraft in the presence of uncertainty. This is the object of the next two chapters. In Chapter 5, we develop a new theory of TMPC for system dynamics in DC form, and in Chapter 6, we apply that new algorithm to robust control of tiltwing VTOL aircraft subject to wind gusts.

5

Difference of convex functions in robust tube MPC

Contents

5.1 Introduction

In the previous chapter, a new robust optimisation algorithm based on successive linearisation and difference of convex (DC) functions was introduced and applied to an urban air mobility scenario. The method exploits tight bounds on the convex linearisation error of the dynamics to compute optimal trajectories for the transition of vertical take-off and landing (VTOL) aircraft. The solution is

robust to model uncertainty introduced by linearisation and can be obtained with predictable computational effort. We noted that this novel method could form the basis of multi-stage optimal control algorithms.

The object of this chapter is to leverage this approach in the framework of robust model predictive control (MPC). We propose a tube-based MPC (TMPC) algorithm for nonlinear systems whose dynamics can be expressed in DC form. The approach exploits the convexity properties of the system model to derive convex conditions that govern the evolution of robust tubes bounding predicted trajectories. These tubes allow an upper bound on a performance cost to be minimised subject to state and control constraints as a convex program, the solution of which can be used to update an estimate of the optimal state and control trajectories. This process is the basis of an iteration that solves a sequence of convex programs at each discrete time step. We show that the algorithm is recursively feasible, converges asymptotically to a fixed point of the iteration and ensures closed loop stability. The algorithm can be terminated after any number of iterations without affecting stability or constraint satisfaction. A case study is presented to illustrate an application of the algorithm. An extension of the algorithm to the case of additive disturbance is also presented. We finally note that the approach is not limited to DC systems but can easily be generalised to systems whose dynamics are twice continuously differentiable and we present a series of techniques to obtain a DC decomposition for such a system.

The chapter is organised as follows. We start by summarising related work in the literature in Section 5.2. We discuss a precursor to the present approach, namely TMPC based on successive linearisation, in Section 5.3. Section 5.4 presents the proposed new approach (DC-TMPC) for DC systems. The convergence, stability and feasibility properties of the algorithm are discussed in Section 5.5. To illustrate the approach, we describe its application to the problem of regulating fluid in a pair of coupled tanks in Section 5.6. We finally discuss extensions of the algorithm to more general nonlinear systems and to systems subject to additive disturbance.

Notation: The value of a variable $x \in \mathbb{R}^{n_x}$ at the nth discrete time step is denoted $x[n]$, and a sequence predicted at discrete time n is denoted

$\{x_k\}_{k=0}^{N-1} := \{x_0, \ldots, x_{N-1}\}$, where x_k is the predicted value of $x[n+k]$. A symmetric matrix $P \in \mathbb{R}^{q\times q}$ is positive definite (resp. semidefinite) if $P \succ 0$ ($P \succeq 0$), and the weighted Euclidean norm is denoted $\|x\|_P := (x^\top P x)^{1/2}$. For $K \in \mathbb{R}^{p\times q}$ and sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^q$ we denote $\mathcal{A} \oplus \mathcal{B} := \{a + b : a \in \mathcal{A},\, b \in \mathcal{B}\}$, $K\mathcal{A} := \{Ka : a \in \mathcal{A}\}$, and $\|\mathcal{A}\|_P := \max_{a\in\mathcal{A}} \|a\|_P$. The lth element of a function $f : \mathbb{R}^p \to \mathbb{R}^q$ is denoted $[f]_l$ and f is called a DC function (or simply DC) if $f = g - h$, where $g, h : \mathbb{R}^p \to \mathbb{R}^q$ are such that $[g]_l, [h]_l$ are convex functions for $l = 1, \ldots, q$. By a slight abuse of notation, optimisation operations for vector-valued functions are intended componentwise, e.g. if f is vector-valued, by $\max_x f(x)$ we mean: $\max_x [f(x)]_l$, $\forall l$.

5.2 Related work

A common strategy in nonlinear TMPC is to treat the nonlinearity in the dynamics as a bounded disturbance [83, 101–103]. In [83], successive linear approximations around predicted trajectories are used to obtain an MPC law for nonlinear systems with differentiable dynamics. The proposed algorithm uses bounds on linearisation errors to construct an ellipsoidal tube and an associated control law. The controller achieves rejection of the linearisation error and stabilisation of the nonlinear system, and the approach has guarantees of recursive feasibility and asymptotic stability.

These strategies rely for robustness on bounds on linearisation errors, which are either computed using fixed prior bounds on predicted future states, and are therefore conservative, or otherwise may present computational difficulties. In this work we show that tighter bounds can be obtained under certain convexity assumptions. We consider systems that can be expressed as a difference of convex functions [127, 133]. The convexity properties of these systems considerably simplify the problem of determining bounds on model states that are suitable for defining predicted state tubes. We derive convex conditions for the evolution of tubes bounding perturbations on future predicted trajectories. These conditions form the basis of an iteration that solves a sequence of convex programs to minimise a bound on predicted cost subject to robust constraint satisfaction.

5.3 Successive linearisation tube MPC

Consider the following discrete-time dynamical system

$$x[n+1] = f(x[n], u[n]). \tag{5.1}$$

Here $x \in \mathcal{X}$, $u \in \mathcal{U}$ are state and control variables and $\mathcal{X} \subset \mathbb{R}^{n_x}$, $\mathcal{U} \subset \mathbb{R}^{n_u}$ are compact constraint sets. The goal is to regulate $(x[n], u[n])$ around a reference $(x^r[n], u^r[n])$ (where $x^r[n+1] = f(x^r[n], u^r[n])$ and $(x^r[n], u^r[n]) \in \mathcal{X} \times \mathcal{U}$), while minimising a quadratic cost. To simplify presentation we discuss here the case of a constant reference (x^r, u^r).

Successive linearisation MPC addresses the optimal control problem to be solved online at each discrete time step by considering perturbations of previously computed predicted trajectories of the system. This approach repeatedly linearises the model (5.1) around state and control trajectories, defined at time n by $\mathbf{x}^\circ = \{x_k^\circ\}_{k=0}^N$, $\mathbf{u}^\circ = \{u_k^\circ\}_{k=0}^{N-1}$, with $x_0^\circ = x_0 = x[n]$ and $x_{k+1}^\circ = f(x_k^\circ, u_k^\circ)$ for $k = 0, \dots, N-1$. In [83] the linearisation errors are treated as unknown bounded disturbances and a robustly stabilising two-degree of freedom controller is designed for the linearised dynamics by solving a sequence of convex optimisation problems.

Defining the predicted state and control perturbations as $s_k = x_k - x_k^\circ$ and $v_k = u_k - u_k^\circ$ respectively, we construct a tube (a sequence of sets) $\boldsymbol{\mathcal{S}} = \{\mathcal{S}_k\}_{k=0}^N$ so that $s_k \in \mathcal{S}_k$ for all k. The perturbed state satisfies $s_0 = 0$ and

$$s_{k+1} = A_k s_k + B_k v_k + g_k,$$

for $k = 0, \dots, N-1$, where $A_k = \frac{\partial f}{\partial x}(x_k^\circ, u_k^\circ)$, $B_k = \frac{\partial f}{\partial u}(x_k^\circ, u_k^\circ)$, and g_k is the remainder term in the first-order Taylor expansion of f about (x_k°, u_k°). For the Jacobian linearisation to exist, the following assumption was implied

Assumption 2. *f is differentiable on $\mathcal{X} \times \mathcal{U}$.*

The predicted control perturbation sequence is parameterised with a feedforward term c_k and a linear feedback term $K_k s_k$, for $k = 0, \dots, N-1$,

$$v_k = c_k + K_k s_k. \tag{5.2}$$

Defining $\Phi_k = A_k + B_k K_k$ and given bounds $g_k \in \mathcal{G}_k$ on linearisation errors, the sets $\mathcal{S}_k$ defining the tube cross-sections therefore satisfy $\mathcal{S}_0 = \{0\}$ and, for $k = 0, \ldots, N-1$,

$$\mathcal{S}_{k+1} \supseteq \Phi_k \mathcal{S}_k \oplus \{B_k c_k\} \oplus \mathcal{G}_k. \tag{5.3}$$

The tube $\boldsymbol{\mathcal{S}}$ and the feedforward sequence $\mathbf{c} = \{c_k\}_{k=0}^{N-1}$ are variables in an optimal tracking control problem with a worst-case cost defined by

$$\begin{aligned} J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ) = \sum_{k=0}^{N-1} \Big[&\|\{x_k^\circ - x^r\} \oplus \mathcal{S}_k\|_Q^2 \\ &+ \|\{u_k^\circ + c_k - u^r\} \oplus K_k \mathcal{S}_k\|_R^2 \Big] + \|\{x_N^\circ - x^r\} \oplus \mathcal{S}_N\|_{\hat{Q}}^2 \end{aligned} \tag{5.4}$$

where $Q \succ 0$, $R \succ 0$. To ensure convergence[1], the terminal weighting matrix $\hat{Q} \succ 0$ is chosen so that, for all $x \in \hat{\mathcal{X}}$,

$$\|x - x^r\|_{\hat{Q}}^2 \geq \|f(x, \hat{K}(x - x^r) + u^r) - x^r\|_{\hat{Q}}^2 + \|x - x^r\|_Q^2 + \|\hat{K}(x - x^r)\|_R^2, \tag{5.5}$$

where the terminal set $\hat{\mathcal{X}}$ and feedback gain $\hat{K}$ satisfy

$$\hat{\mathcal{X}} \subseteq \mathcal{X}, \quad \hat{K}\hat{\mathcal{X}} \oplus \{u^r - \hat{K}x^r\} \subseteq \mathcal{U}. \tag{5.6}$$

The terminal set can be defined, for example, as

$$\hat{\mathcal{X}} = \{x : \|x\|_{\hat{Q}}^2 \leq \hat{\gamma}\} \tag{5.7}$$

and the terminal parameters $\hat{K}$, $\hat{Q}$ and $\hat{\gamma}$ can be computed offline by solving a semidefinite program (cf. Appendix B). The gains $\{K_k\}_{k=0}^{N-1}$ in equation (5.2) can be computed, for example, using a dynamic programming (DP) recursion initialised with $P_N = \hat{Q}$, and defined for $k = N-1, \ldots, 0$ by

$$\begin{aligned} \Delta_k &= B_k^\top P_{k+1} B_k + R \\ K_k &= -\Delta_k^{-1} B_k^\top P_{k+1} A_k \\ P_k &= Q + A_k^\top P_{k+1} A_k - A_k^\top P_{k+1} B_k \Delta_k^{-1} B_k^\top P_{k+1} A_k \end{aligned} \tag{5.8}$$

[1]Equation (5.5) can be seen as a descent property for the Lyapunov function $V(x - x^r) = ||x - x^r||_{\hat{Q}}^2$.

The MPC optimisation at time n is initialised with a feasible predicted trajectory $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ of (5.1) and the following optimisation problem in variables $\mathbf{c}$, $\boldsymbol{\mathcal{S}}$ is solved sequentially

$$\begin{aligned}\min_{\mathbf{c},\boldsymbol{\mathcal{S}}}\ & J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ) \text{ subject to, } \forall k \in \{0, \dots, N-1\},\\ & \Phi_k \mathcal{S}_k \oplus \{B_k c_k\} \oplus \mathcal{G}_k \subseteq \mathcal{S}_{k+1}\\ & \{x_k^\circ\} \oplus \mathcal{S}_k \subseteq \mathcal{X}\\ & \{u_k^\circ + c_k\} \oplus K_k \mathcal{S}_k \subseteq \mathcal{U}\\ & \|\{x_N^\circ - x^r\} \oplus \mathcal{S}_N\|_{\hat{Q}}^2 \leq \hat{\gamma}\\ & \mathcal{S}_0 = \{0\}.\end{aligned} \tag{5.9}$$

The solution of (5.9) for $\mathbf{c}$ is used to update $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ by setting

$$s_0 \leftarrow 0, \tag{5.10a}$$

$$u_k^\circ \leftarrow u_k^\circ + c_k + K_k s_k, \tag{5.10b}$$

$$s_{k+1} \leftarrow f(x_k^\circ, u_k^\circ) - x_{k+1}^\circ, \tag{5.10c}$$

$$x_{k+1}^\circ \leftarrow f(x_k^\circ, u_k^\circ), \tag{5.10d}$$

for $k = 0, \dots, N-1$, and the process (5.9)-(5.10) is repeated until $\|\mathbf{c}\|^2 := \sum_{k=0}^{N-1} \|c_k\|^2$ has converged to a sufficiently small value or the maximum number of iterations is reached. The control law is then implemented as

$$u[n] = u_0^\circ. \tag{5.11}$$

At time $n+1$ we set $x_0^\circ \leftarrow x[n+1]$ and initialise a new iteration with $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ defined, for $k = 0, \dots, N-1$, by

$$u_k^\circ \leftarrow u_{k+1}^\circ, \tag{5.12a}$$

$$x_{k+1}^\circ \leftarrow f(x_k^\circ, u_k^\circ), \tag{5.12b}$$

$$u_{N-1}^\circ \leftarrow \hat{K}(x_{N-1}^\circ - x^r) + u^r, \tag{5.12c}$$

$$x_N^\circ \leftarrow f(x_{N-1}^\circ, u_{N-1}^\circ). \tag{5.12d}$$

These definitions ensure recursive feasibility of the scheme . In addition, the optimal value of $J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ)$ is non-increasing at successive iterations, implying that

the iteration converges asymptotically: $\mathbf{c} \to 0$ and $\boldsymbol{\mathcal{S}} \to \{\{0\}, \ldots, \{0\}\}$, and also non-increasing at successive time steps $n = 0, 1, \ldots$, implying that $(x, u) = (x^r, u^r)$ is an asymptotically stable equilibrium (see e.g. [83], Theorems 7 and 8 for details).

Despite its attractive theoretical properties, the successive linearisation approach relies on bounds on the linearisation error $\mathcal{G}_k$, which are difficult to compute in general and might result in overly conservative bounds if no further assumptions are made on f. Moreover, the algorithm outlined in this section does not ensure convergence to a solution that satisfies the first order optimality conditions for the problem of minimizing $\sum_{k=0}^{N-1}[\|x_k - x^r\|_Q^2 + \|u_k - u^r\|_R^2] + \|x_N - x^r\|_{\hat{Q}}^2$ subject to $x_{k+1} = f(x_k, u_k)$, $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$ since $\mathcal{G}_k$ may not be tight. To overcome these problems, we propose a method for systems (5.1) in which f is a DC function to obtain tight bounds on linearisation errors.

5.4 Successive convex programming tube MPC for DC systems

To derive a control algorithm with reduced computation and improved performance, we strengthen our assumptions on model (5.1) as follows.

Assumption 3. *f is DC on $\mathcal{X} \times \mathcal{U}$ and $\mathcal{X}$,$\mathcal{U}$ are convex sets.*

Under Assumption 3 the model (5.1) can be expressed

$$x[n+1] = f_1(x[n], u[n]) - f_2(x[n], u[n]), \tag{5.13}$$

where each component of the vector-valued functions f_1, f_2 is a convex function. Note that (5.13) includes systems in which f in (5.1) is either convex or concave. More generally, any twice-continuously-differentiable function is DC (see Appendix A, Theorem 13) and any continuous function can be approximated arbitrarily closely by a DC function on a compact convex set [127, 133], which extends considerably the scope of the present method. We have presented a method to compute a DC decomposition for a class $\mathcal{C}^2$ function of one variable using polynomial approximation

techniques in Chapter 4, and we discuss other more general techniques at the end of this chapter, see Section 5.7.2.

Given a predicted trajectory $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ of (5.1), the perturbed variables $s_k = x_k - x_k^\circ$, $v_k = u_k - u_k^\circ$ are governed by $s_0 = 0$ and, for $k = 0, \dots, N-1$,

$$s_{k+1} = A_{1,k}s_k + B_{1,k}v_k + g_{1,k} - (A_{2,k}s_k + B_{2,k}v_k + g_{2,k}),$$

where for $i = 1, 2$, $A_{i,k} = \frac{\partial f_i}{\partial x}(x_k^\circ, u_k^\circ)$, $B_{i,k} = \frac{\partial f_i}{\partial u}(x_k^\circ, u_k^\circ)$, and $g_{i,k}$ represents the error in the Jacobian linear approximation of f_i around (x_k°, u_k°). Using the control law (5.2), defining $\Phi_{i,k} = A_{i,k} + B_{i,k}K_k$, $i = 1, 2$, and assuming bounds $g_{i,k} \in \mathcal{G}_{i,k}$, $i = 1, 2$, we obtain a set of constraints to be satisfied by the tube cross-sections $\mathcal{S}_k$ containing s_k:

$$\mathcal{S}_{k+1} \supseteq (\Phi_{1,k} - \Phi_{2,k})\mathcal{S}_k \oplus \{(B_{1,k} - B_{2,k})c_k\} \oplus \mathcal{G}_{1,k} \oplus -\mathcal{G}_{2,k}. \tag{5.14}$$

We now characterise the sets $\mathcal{S}_k$ and $\mathcal{G}_{i,k}$ to obtain a workable formulation of these constraints. Various parameterisations of the sets $\mathcal{S}_k$ are possible: polytopic, ellipsoidal, homothetic (e.g. [69]). In this work, for simplicity, we define a polytopic tube cross-section in terms of elementwise bounds:

$$\mathcal{S}_k = \{(s_1, \dots, s_{n_x}) \in \mathbb{R}^{n_x} : \underline{s}_{k,j} \leq s_j \leq \bar{s}_{k,j},\ j = 1, \dots, n_x\}.$$

Considering $g_{1,k}$, $g_{2,k}$ as functions of s_k and c_k:

$$g_{i,k} = f_i(x_k^\circ + s_k, u_k^\circ + K_k s_k + c_k) - f_i(x_k^\circ, u_k^\circ) - \Phi_{i,k}s_k - B_{i,k}c_k, \tag{5.15}$$

the convexity of $f_i(x, u)$ for $(x, u) \in \mathcal{X} \times \mathcal{U}$ implies that $g_{i,k}(s, c)$ is convex[2] for all (s, c) such that $(x^\circ + s, u^\circ + Ks + c)$ lies in $\mathcal{X} \times \mathcal{U}$. It follows that[3], for $i = 1, 2$ and $l = 1, \dots, n_x$,

$$\min_{s_k \in \mathcal{S}_k} g_{i,k}(s_k, c_k) = 0,$$
$$\max_{s_k \in \mathcal{S}_k} g_{i,k}(s_k, c_k) = \max_{\hat{s}_k \in \mathcal{V}(\mathcal{S}_k)} g_{i,k}(\hat{s}_k, c_k),$$

[2] Appendix A, Theorem 14.
[3] Ibid., Theorem 16.

where $\mathcal{V}(\mathcal{S}_k)$ denotes the set of vertices of $\mathcal{S}_k$. The tube constraint (5.14) is therefore ensured by the convex inequalities

$$\begin{aligned}\underline{s}_{k+1} &\leq \min_{\hat{s}_k \in \mathcal{V}(\mathcal{S}_k)} \{(\Phi_{1,k}-\Phi_{2,k})\hat{s}_k + (B_{1,k}-B_{2,k})c_k - g_{2,k}(\hat{s}_k)\},\\ \overline{s}_{k+1} &\geq \max_{\hat{s}_k \in \mathcal{V}(\mathcal{S}_k)} \{(\Phi_{1,k}-\Phi_{2,k})\hat{s}_k + (B_{1,k}-B_{2,k})c_k + g_{1,k}(\hat{s}_k)\},\end{aligned}$$

where the maximum and minimum operations are over the vertices of the set (and not over the whole set), which reduces considerably the computational burden associated with their evaluation. If no prior knowledge of their linear coefficients is exploited, each of these inequalities is equivalent to 2^{n_x} inequalities in $\mathbf{c}$ and the parameters $\{\underline{s}_k, \overline{s}_k\}_{k=0}^{N-1}$ that define $\boldsymbol{\mathcal{S}}$ as follows, $\forall \hat{s}_k \in \mathcal{V}(\mathcal{S}_k)$

$$\begin{aligned}\underline{s}_{k+1} &\leq \Phi_{1,k}\hat{s}_k + B_{1,k}c_k - (f_2(x_k^\circ + \hat{s}_k, u_k^\circ + K_k\hat{s}_k + c_k) - f_2(x_k^\circ, u_k^\circ)),\\ \overline{s}_{k+1} &\geq f_1(x_k^\circ + \hat{s}_k, u_k^\circ + K_k\hat{s}_k + c_k) - f_1(x_k^\circ, u_k^\circ) - (\Phi_{2,k}\hat{s}_k + B_{2,k}c_k),\end{aligned}$$

where we substituted for the linearisation errors as defined in equation (5.15). Crucially, the resulting inequalities are convex, and sequences $\mathbf{c}$, $\boldsymbol{\mathcal{S}}$ can be computed for given $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ by solving a convex problem

$$\begin{aligned}&\min_{\mathbf{c},\boldsymbol{\mathcal{S}}} \; J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ)\\ &\text{subject to, } \forall k \in \{0, \dots, N-1\},\ \forall \hat{s}_k \in \mathcal{V}(\mathcal{S}_k):\\ &\underline{s}_{k+1} \leq \Phi_{1,k}\hat{s}_k + B_{1,k}c_k - (f_2(x_k^\circ + \hat{s}_k, u_k^\circ + K_k\hat{s}_k + c_k) - f_2(x_k^\circ, u_k^\circ))\\ &\overline{s}_{k+1} \geq f_1(x_k^\circ + \hat{s}_k, u_k^\circ + K_k\hat{s}_k + c_k) - f_1(x_k^\circ, u_k^\circ) - (\Phi_{2,k}\hat{s}_k + B_{2,k}c_k)\\ &x_k^\circ + \hat{s}_k \subseteq \mathcal{X}\\ &u_k^\circ + K_k\hat{s}_k + c_k \subseteq \mathcal{U}\\ &\|x_N^\circ + \hat{s}_N - x^r\|_{\hat{Q}}^2 \leq \hat{\gamma},\ \forall \hat{s}_N \in \mathcal{V}(\mathcal{S}_N)\\ &\underline{s}_0 = \overline{s}_0 = 0.\end{aligned} \tag{5.16}$$

Problem (5.16) is solved repeatedly and $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ is updated according to (5.10a-d), until $\|\mathbf{c}\|$ is sufficiently small or the maximum number of iterations is reached. The control law is defined as (5.11), and at the subsequent discrete time instant we set $x_0^\circ \leftarrow x[n+1]$, and redefine $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ using (5.12a-d). The tube-based MPC (TMPC) strategy for DC systems is summarised in Algorithm 2. We refer to this as DC-TMPC in what follows.

Remark 1. *The number of inequalities defining the tube grows exponentially with the number of states n_x (e.g. there are $|\mathcal{V}(\mathcal{S})| = 2^{n_x}$ vertices associated with sets defined by elementwise bounds). That growth is further exacerbated if higher dimensional polytopes are used. In practice, most systems have a reasonable number of states. Moreover, prior knowledge of the state coefficients appearing linearly in these inequalities can potentially be exploited to reduce the number of inequalities. Finally, if computational burden is an issue, simplex sets can be used that scale linearly with the number of states: only $n_x + 1$ inequalities needed.*

Remark 2. *The objective in problem (5.16) is a worst case objective that normally involves maximum operations over the sets $\boldsymbol{\mathcal{S}}$ as follows*

$$J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ) = \sum_{k=0}^{N-1} \max_{s_k \in \mathcal{S}_k} \left[||x_k^\circ + s_k - x^r||_Q^2 + ||u_k^\circ + c_k + K_k s_k - u^r||_R^2 \right] + \max_{s_N \in \mathcal{S}_N} ||x_N^\circ + s_N - x^r||_{\hat{Q}}^2.$$

However, by convexity of the norm and since the sets $\boldsymbol{\mathcal{S}}$ are polytopes, these operations can be replaced by a discrete number of evaluations/comparisons of the norm functions at the vertices of the sets $\boldsymbol{\mathcal{S}}$, which simplifies considerably the problem. This is because a convex function defined on a polytope achieves its maximum at one of the vertices (Appendix A, Lemma 15), so that maximisation over a continuous space can be replaced by a discrete search.

Finally, evaluation of the worst case cost can be achieved in practice by redefining it using slack variables as follows

$$J = ||\theta_N||_{\hat{Q}}^2 + \sum_{k=0}^{N-1} \Big[||\theta_k||_Q^2 + ||\chi_k||_R^2 \Big],$$

subject to

$$\theta_k \geq |x_k^\circ + \hat{s}_k - x^r|, \quad \forall k \in \{0, \dots, N-1\}, \quad \forall \hat{s}_k \in \mathcal{V}(\mathcal{S}_k),$$
$$\chi_k \geq |u_k^\circ + c_k + K_k \hat{s}_k - u^r|, \quad \forall k \in \{0, \dots, N-1\}, \quad \forall \hat{s}_k \in \mathcal{V}(\mathcal{S}_k),$$
$$\theta_N \geq |x_N^\circ + \hat{s}_N - x^r|, \quad \forall \hat{s}_N \in \mathcal{V}(\mathcal{S}_N),$$

where $|x|$ represents componentwise absolute value when applied to a vector $x \in \mathbb{R}^{n_x}$, i.e. $\Big|[x]_l\Big|$, $\forall l \in \{1, \dots, n_x\}$.

Remark 3. *The DC-TMPC strategy requires (in line 7 of Algorithm 2) knowledge of a feasible predicted trajectory at time $n = 0$, namely a sequence $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ satisfying $(x_k^\circ, u_k^\circ) \in \mathcal{X} \times \mathcal{U}$, $k = 0, \ldots, N-1$ and $\|x_N^\circ - x^r\|_{\hat{Q}}^2 \leq \hat{\gamma}$. An iterative approach to determining a feasible trajectory can be constructed by relaxing some of the constraints of (5.16) and performing successive optimisations to minimise constraint violations. For example, given a trajectory $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ of (5.1) that satisfies $(x_k^\circ, u_k^\circ) \in \mathcal{X} \times \mathcal{U}$, $k = 0, \ldots, N-1$, but not the terminal constraint $\|x_N^\circ - x^r\|_{\hat{Q}}^2 \leq \hat{\gamma}$, a suitable iteration can be constructed by replacing $\hat{\gamma}$ in (5.16) with an optimisation variable γ and replacing the objective of (5.16) with γ. Repeatedly solving this convex program and updating $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ using (5.10a-d) will generate a convergent sequence of predicted trajectories with non-increasing values of $\|x_N^\circ - x^r\|_{\hat{Q}}^2$, which can be terminated when $\gamma \leq \hat{\gamma}$ is achieved.*

5.5 Feasibility, convergence and stability

This section establishes some important properties of DC-TMPC. We first show that $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ is a feasible predicted trajectory at every iteration of Algorithm 2 and at all times $n \geq 0$. This ensures firstly that problem (5.16) in line 18 is recursively feasible, and second that the while loop (lines 12-22) can be terminated after any number of iterations without compromising the stability of the MPC law, i.e. the approach allows *early termination* of the MPC optimisation.

Proposition 3 (Feasibility)**.** *Assuming initial feasibility at time $n = 0$ and iteration $j = 0$, problem (5.16) in Algorithm 2 is feasible at all times $n > 0$ and all iterations $j > 0$.*

Proof. If $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ is a feasible trajectory, namely a trajectory of (5.1) satisfying $(x_k^\circ, u_k^\circ) \in \mathcal{X} \times \mathcal{U}$, $k = 0, \ldots, N-1$ and $\|x_N^\circ - x^r\|_{\hat{Q}}^2 \leq \hat{\gamma}$, then problem (5.16) is feasible since $(\mathbf{c}, \boldsymbol{\mathcal{S}}) = (0, \{\{0\}, \ldots, \{0\}\})$ trivially satisfies all constraints of (5.16). Furthermore, line 7 of Algorithm 2 requires that $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ is a feasible trajectory at $n = 0$, and thus ensures that (5.16) in line 18 is feasible for $n = 0$, $j = 0$.

Algorithm 2: DC-TMPC algorithm.

1 **Initialisation**
2 | Compute $\hat{K}$, $\hat{Q}$, $\hat{\gamma}$ satisfying (5.5), (5.6) and (5.7)
3 **end**
4 **Online at times** $n = 0, 1, \ldots$
5 | Set $x_0^\circ \leftarrow x[n]$
6 | **if** $n = 0$ **then**
7 | | Find $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ such that $x_{k+1}^\circ = f(x_k^\circ, u_k^\circ)$, $(x_k^\circ, u_k^\circ) \in \mathcal{X} \times \mathcal{U}$ for $k = 0, \ldots, N-1$, and $\|x_N^\circ - x^r\|_{\hat{Q}}^2 \leq \hat{\gamma}$
8 | **else**
9 | | Compute $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ using (5.12a-d)
10 | **end**
11 | Set $\|\mathbf{c}\| \leftarrow \infty$, $j \leftarrow 0$
12 | **while** $\|\mathbf{c}\| >$ *tolerance* & $j <$ *maxIters* **do**
13 | | **for** $k \leftarrow N-1$ **to** 0 **do**
14 | | | $A_{i,k} \leftarrow \frac{\partial f_i}{\partial x}(x_k^\circ, u_k^\circ)$, $B_{i,k} \leftarrow \frac{\partial f_i}{\partial u}(x_k^\circ, u_k^\circ)$, $i = 1, 2$
15 | | | Compute K_k in (5.8) with $A_k = A_{1,k} - A_{2,k}$, $B_k = B_{1,k} - B_{2,k}$
16 | | | Set $\Phi_{i,k} \leftarrow A_{i,k} + B_{i,k} K_k$
17 | | **end**
18 | | Solve (5.16) to find the optimal $\mathbf{c}$ and $\boldsymbol{\mathcal{S}}$
19 | | Update $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ using (5.10a-d)
20 | | Update the objective: $J^{j,n} \leftarrow J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ)$
21 | | Update the iteration counter: $j \leftarrow j + 1$
22 | **end**
23 | Update the control input: $u[n] \leftarrow u_0^\circ$
24 **end**

Therefore $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ computed in lines 9 and 19 of Algorithm 2 are necessarily feasible trajectories for all $n \geq 0$ and $j \geq 0$. □

We next show that, at a given time step n, the optimal values of the objective function for problem (5.16) computed at successive iterations of Algorithm 2 are non-increasing.

Lemma 4. *The cost $J^{j,n}$ in line 20 of Algorithm 2 satisfies $J^{j+1,n} \leq J^{j,n}$ for all $j \geq 0$.*

Proof. This is a consequence of the recursive feasibility of problem (5.16) in line 18 of Algorithm 2 as demonstrated by Proposition 3. Specifically, $J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ)$ is the maximum of a quadratic cost evaluated over the predicted tube, so the feasible

suboptimal solution $(\mathbf{c}, \boldsymbol{\mathcal{S}}) = (0, \{\{0\}, \ldots, \{0\}\})$ at iteration $j+1$ must give a cost no greater than $J^{j,n}$. Therefore the optimal value at $j+1$ satisfies $J^{j+1,n} \leq J^{j,n}$. □

Lemma 4 implies that $J^{j,n}$ converges to a limit as $j \to \infty$, and this is the basis of the following result, which implies convergence of the linearisation error to zero.

Proposition 5 (Convergence)**.** *The solution of (5.16) in Algorithm 2 at successive iterations at a given time step n satisfies $(\mathbf{c}, \boldsymbol{\mathcal{S}}) \to (0, \{\{0\}, \ldots, \{0\}\})$ as $j \to \infty$.*

Proof. From $J^{j,n} \geq 0$ and $J^{j+1,n} \leq J^{j,n}$ for all j, it follows that $J^{j,n}$ converges to a limit as $j \to \infty$. Since $Q \succ 0$ and $R \succ 0$, this implies that $\mathbf{c} \to 0$ and s_k, $k = 0, \ldots, N-1$, in the update (5.10a-d) in line 19 must also converge to zero. Furthermore, since $(\mathbf{c}, \boldsymbol{\mathcal{S}})$ minimizes the worst-case cost evaluated for the predicted tube, the optimal solutions of (5.16) must satisfy $\boldsymbol{\mathcal{S}} \to \{\{0\}, \ldots, \{0\}\}$ as $j \to \infty$. □

We finally show that under the DC-TMPC law, $x = x^r$ is asymptotically stable.

Theorem 6 (Asymptotic stability)**.** *For the system (5.1) controlled by DC-TMPC, $x = x^r$ is an asymptotically stable equilibrium with region of attraction equal to the set of initial states $x[0]$ for which a feasible initial trajectory $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ exists in line 7 of Algorithm 2.*

Proof. Let $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ be the trajectory computed in line 9 of Algorithm 2 at $n+1$, for any initial state such that line 7 is feasible. Then, setting $(\mathbf{c}, \boldsymbol{\mathcal{S}}) = (0, \{\{0\}, \ldots, \{0\}\})$, and using $\mathcal{S}_0 = \{0\}$ and the property (5.5) of $\hat{Q}$, we obtain

$$J(\mathbf{c}, \boldsymbol{\mathcal{S}}, \mathbf{x}^\circ, \mathbf{u}^\circ) \leq J^{j_n,n} - \|x[n] - x^r\|_Q^2 - \|u[n] - u^r\|_R^2,$$

where j_n is the final iteration count of Algorithm 2 at time n. The optimal solutions of (5.16) at time $n+1$ therefore give

$$J^{j_{n+1},n+1} \leq J^{0,n+1} \leq J^{j_n,n} - \|x[n] - x^r\|_Q^2 - \|u[n] - u^r\|_R^2.$$

Furthermore $J^{j,n}$ is a positive definite function of $x[n] - x^r$ since $Q \succ 0$, $R \succ 0$, and Lyapunov's direct method therefore implies that $x = x^r$ is asymptotically stable. □

5.6 Case study

This section presents a simple case study to illustrate the use of the DC-TMPC algorithm and demonstrate empirically its properties. The case study is based on the problem of fluid regulation for the coupled tank system, which not only is a famous benchmark in control theory, but also has the property of being already in the required DC form. Application of the DC-TMPC algorithm to the urban air mobility scenario this thesis is concerned with is deferred to the next chapter as it is a more complex problem requiring a DC decomposition of the dynamics and inclusion of external (wind) disturbance.

Consider the following nonlinear coupled tank model

$$\begin{aligned} x_1[n+1] &= x_1[n] - \delta\frac{A_1}{A}\sqrt{2gx_1[n]} + \delta\frac{k_p}{A}u[n], \\ x_2[n+1] &= x_2[n] - \delta\frac{A_2}{A}\sqrt{2gx_2[n]} + \delta\frac{A_1}{A}\sqrt{2gx_1[n]}, \end{aligned}$$

where x_1, x_2, are the depths of liquid in a pair of connected tanks, A is the tank cross-section area, A_1, A_2 are the outflow orifices areas, g is acceleration due to gravity, and k_p is the pump gain. The problem at hand is to stabilise the system around a reference state $x_k^r = [(A_2/A_1)^2 h_r \quad h_r]^\top$. The model can be written as a difference of convex functions

$$\begin{aligned} f &= \begin{bmatrix} x_1[n] - \delta\frac{A_1}{A}\sqrt{2gx_1[n]} + \delta\frac{k_p}{A}u[n] \\ x_2[n] - \delta\frac{A_2}{A}\sqrt{2gx_2[n]} + \delta\frac{A_1}{A}\sqrt{2gx_1[n]} \end{bmatrix}, \\ &= \underbrace{\begin{bmatrix} x_1[n] - \delta\frac{A_1}{A}\sqrt{2gx_1[n]} + \delta\frac{k_p}{A}u[n] \\ x_2[n] - \delta\frac{A_2}{A}\sqrt{2gx_2[n]} \end{bmatrix}}_{f_1} - \underbrace{\begin{bmatrix} 0 \\ -\delta\frac{A_1}{A}\sqrt{2gx_1[n]} \end{bmatrix}}_{f_2} \end{aligned}$$

Let $(\mathbf{x}_k^\circ, \mathbf{u}_k^\circ)$ be an N-step-ahead predicted trajectory evaluated at time n. Since the model depends linearly on the control input, the perturbations $s_k = x_k - x_k^\circ$ and $v_k = u_k - u_k^\circ$ under the predicted control law (5.2) are governed by the dynamics

$$s_{k+1} = (\Phi_{1,k} - \Phi_{2,k})s_k + (B_{1,k} - B_{2,k})c_k + g_{1,k}(s_k) - g_{2,k}(s_k)$$

where $\Phi_{1,k} = A_{1,k} + B_{1,k}K_k$, $\Phi_{2,k} = A_{2,k} + B_{2,k}K_k$ and

$$A_{1,k} = \begin{bmatrix} 1 - \frac{\delta A_1 g}{A\sqrt{2gx_{1,k}^\circ}} & 0 \\ 0 & 1 - \frac{\delta A_2 g}{A\sqrt{2gx_{2,k}^\circ}} \end{bmatrix},$$

$$A_{2,k} = \begin{bmatrix} 0 & 0 \\ -\frac{\delta A_1 g}{A\sqrt{2g x^{\circ}_{1,k}}} & 0 \end{bmatrix}, \quad B_{1,k} = \begin{bmatrix} \frac{\delta k_p}{A} \\ 0 \end{bmatrix}, \quad B_{2,k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The feedback gains K_k were computed using the dynamic programming recursion (5.8) with $A_k = A_{1,k} - A_{2,k}$ and $B_k = B_{1,k} - B_{2,k}$ and state and input weights Q, R. The terminal parameters $\hat{K}$, $\hat{Q}$ and $\hat{\gamma}$ were obtained by solving the optimisation problem given in Appendix B with square terminal set bounds $\bar{\mathcal{X}} = \{x : |x - x^r| \leq \delta^x\}$, $\bar{\mathcal{U}} = \{u : |u - u^r| \leq \delta^u\}$. Denoting the bounds defining tube cross-sections $\mathcal{S}_k$ as $\underline{s}_k = [\underline{s}_{k,1} \;\; \underline{s}_{k,2}]^\top$, $\overline{s}_k = [\overline{s}_{k,1} \;\; \overline{s}_{k,2}]^\top$, the vertices of $\mathcal{S}_k$ are

$$\mathcal{V}(\mathcal{S}_k) = \left\{ \begin{bmatrix} \underline{s}_{k,1} \\ \underline{s}_{k,2} \end{bmatrix}, \begin{bmatrix} \overline{s}_{k,1} \\ \underline{s}_{k,2} \end{bmatrix}, \begin{bmatrix} \underline{s}_{k,1} \\ \overline{s}_{k,2} \end{bmatrix}, \begin{bmatrix} \overline{s}_{k,1} \\ \overline{s}_{k,2} \end{bmatrix} \right\},$$

and the state and input constraint sets have the form $\mathcal{X} = \{x : \underline{x}[1 \; 1]^\top \leq x \leq \overline{x}[1 \; 1]^\top\}$, $\mathcal{U} = \{u : \underline{u} \leq u \leq \overline{u}\}$. Table 5.1 gives the model parameters.

To apply DC-TMPC to the coupled tank problem, Algorithm 2 was initialised by setting $\mathbf{u}^\circ$ to a constant voltage $u^r = 7.3\ V$, resulting in a feasible initial trajectory $(\mathbf{x}^\circ, \mathbf{u}^\circ)$ for a horizon $N = 50$ and sampling interval $\delta = 1.4\,\text{s}$. Problem (5.16) was solved[4] using the convex optimisation package CVX [123] with solver Mosek [132]. The average computation time required (using a 2.9 GHz dual-core Intel Core i7 processor) at each sampling instant to implement Algorithm 2 was 0.89 s.

Figure 5.1 shows that the system successfully tracks the reference levels and the fluid level in tank 2 is stabilised around a height of 15 cm as required. Note that the overshoot for tank 1 is large, allowing the tanks to fill faster.

The influence of the input penalty R on the response is shown in Figure 5.2. For a large R, the response is slow with an energy efficient control law. By contrast, small values of R yield a more aggressive control with faster responses. Interestingly, the response for $R = 0.02$ makes the state and input inequality constraints active, which demonstrates the capabilities of the algorithm to generate a control command that does not violate constraints.

[4]The code in Python is available at: `https://github.com/martindoff/DC-TMPC`

Convergence of the algorithm is demonstrated empirically in Figure 5.3 which shows the evolution of the first-iteration optimal objective $J^{0,n}$ as a function of the time step n. As expected, the objective decreases at each step.

The phase portrait in Figure 5.4 illustrates convergence of state trajectories and subsequent tightening of the state perturbation bounds for iterations $j = 1, \ldots, 5$ at time step $n = 0$. The sets $\mathcal{S}_k$ forming the cross sections of the tube are shown by black boxes. The sets become progressively tighter as the trajectory converges towards the optimum. The terminal set is represented by a red box, and we observe that all trajectories terminate within this set.

Table 5.1: Coupled Tank Parameters [134].

Parameter	**Symbol**	**Value**	**Units**
Gravity acceleration	g	981	$\mathrm{cm\,s^{-2}}$
Pump gain	k_p	3.3	$\mathrm{cm^3 s^{-1} V^{-1}}$
Tank inside area	A	15.2	$\mathrm{cm^2}$
Outflow orifice areas	A_1, A_2	0.13, 0.14	$\mathrm{cm^2}$
Initial height	$x_1(0), x_2(0)$	0.2, 0.1	cm
Target height	h^r	15	cm
Target voltage	u^r	7.3	V
Input range	$[\underline{u}, \overline{u}]$	$[0, 24]$	V
State range	$[\underline{x}, \overline{x}]$	$[0.1, 30]$	cm
Terminal set size	δ^x, δ^u	$[1.5\ 1.5]^\top, 1.5$	cm, V
Terminal cost	$\hat{Q}$	$\begin{bmatrix} 2.1 & 1.2 \\ 1.2 & 5.9 \end{bmatrix}$	$\mathrm{cm^{-2}}$
Terminal gain	$\hat{K}$	$\begin{bmatrix} -0.8 & -0.6 \end{bmatrix}$	$\mathrm{Vcm^{-1}}$
Terminal bound	$\hat{\gamma}$	4	–
State penalty	Q	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\mathrm{cm^{-2}}$
Input penalty	R	0.2	$\mathrm{V^{-2}}$
Horizon	N	50	–
Time step	δ	1.4	s
Max # of iterations	maxIters	5	–

We now compare the convergence properties of DC-TMPC with the successive linearisation TMPC algorithm in [83] (MPC-2011). As described in Section 5.3, linearisation errors around predicted trajectories are treated as disturbances in MPC-2011. The approach uses state- and control-dependent bounds on these errors, but these are determined assuming a fixed operating region. Hence the approach

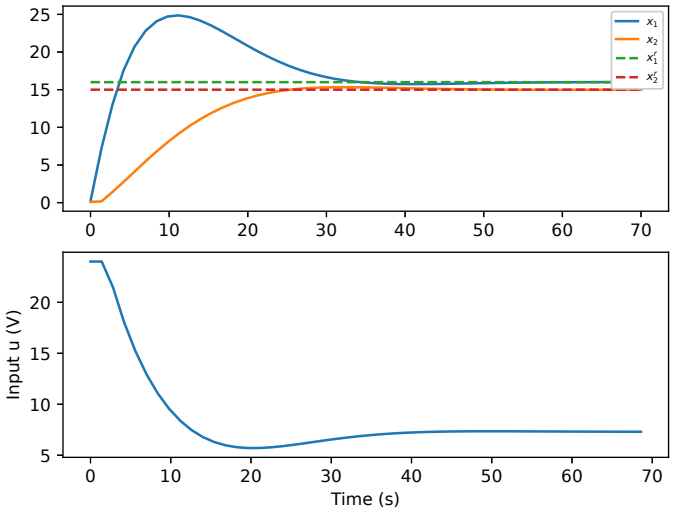


Figure 5.1: State and input trajectories.

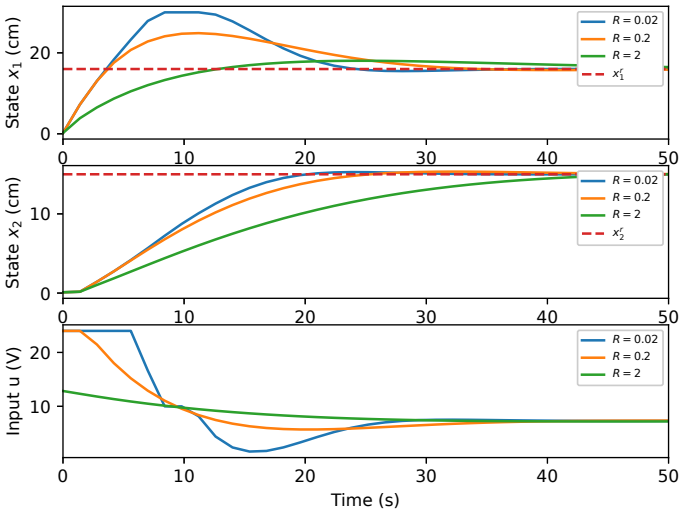


Figure 5.2: Influence of input penalty R on the closed-loop response.

is more conservative than DC-TMPC, which exploits the convex nature of the linearisation errors to find tighter bounds on the state perturbations. As a result, it is expected that DC-TMPC demonstrates faster convergence and a larger set of feasible initial conditions than MPC-2011. To demonstrate this, we apply the MPC-2011 algorithm to the coupled tanks model with the same parameters. For a given terminal set, the range of open loop input voltage allowable for initialising

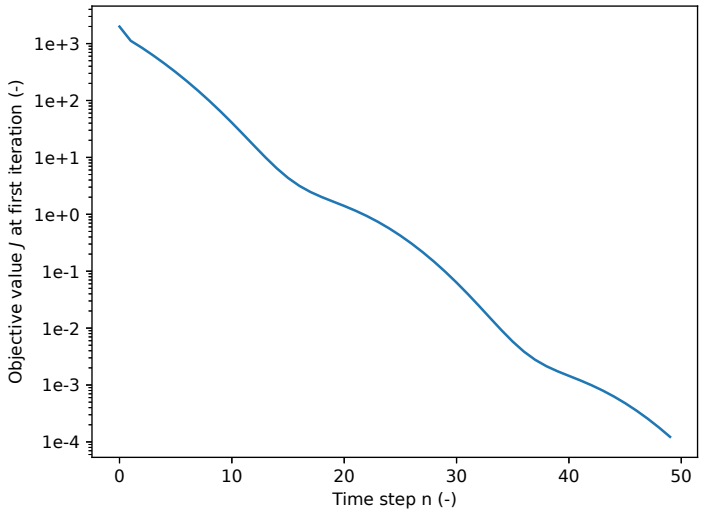


Figure 5.3: Evolution of the objective value at first iteration, $J^{1,n}$.

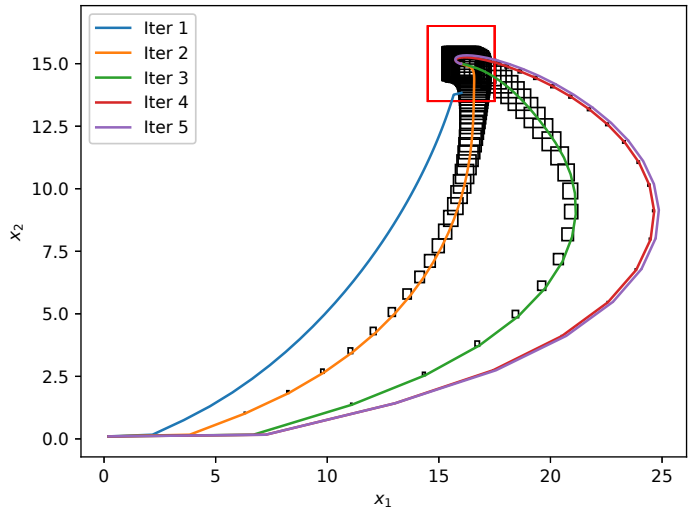


Figure 5.4: Phase portrait at time step $n = 0$ with successive predicted state trajectories, associated bounds (black boxes) and terminal set (red box).

the algorithm with a feasible problem was found to be $[6.1, 9.3]$ V for DC-TMPC, while it was limited between $[7.2, 7.8]$ V for MPC-2011, showing a smaller feasible initial conditions set. This demonstrates the relative conservativeness of the state perturbation bounds in MPC-2011 over DC-TMPC, as expected. Finally, the faster convergence of DC-TMPC is shown in Figure 5.5, which compares the evolution of the objective value for both algorithms at the first time step, $J^{j,0}$, $j = 1, \ldots, 5$.

This demonstrate the superiority of DC-TMPC over the tube-based MPC-2011 algorithm with successive linearisations.

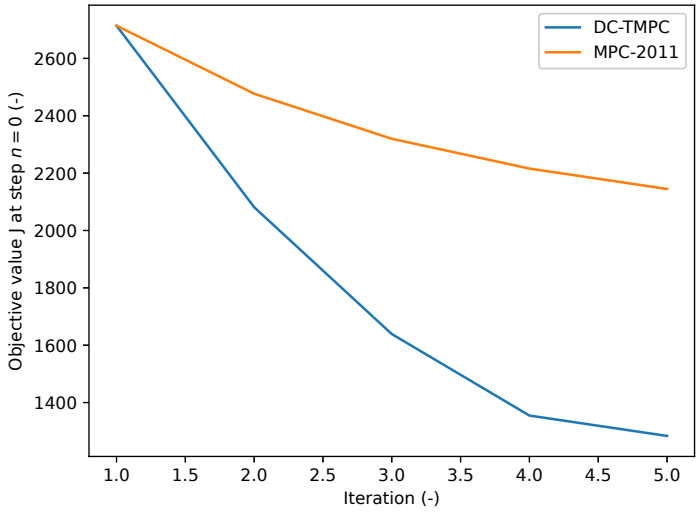


Figure 5.5: Comparison of the objective value $J^{j,0}$ for both algorithms.

5.7 Extensions

We now discuss extensions of the method to i) systems subject to additive uncertainty; ii) systems whose dynamics are twice continuously differentiable.

5.7.1 Additive uncertainty

The DC-TMPC algorithm presented in this chapter can also be applied (with minor adjustments) to systems subject to additive uncertainty

$$x[n+1] = f_1(x[n], u[n]) - f_2(x[n], u[n]) + w[n], \quad w[n] \in \mathcal{W}, \tag{5.17}$$

where f_1, f_2 are convex, w is a bounded disturbance and $\mathcal{W} \subset \mathbb{R}^{n_x}$ is a known compact set.

Since the system is perturbed by additive disturbances whose future magnitude is uncertain, the trajectories of system (5.17) under a known control law are uncertain. Therefore, the tube bounding predicted states cannot converge to a single trajectory

as in the undisturbed case, and the predicted trajectories used to linearise the dynamics should now be replaced by a tube.

Required adjustments to the undisturbed case were proposed in [135] for convex systems. We now extend the theory to DC systems with the following modifications:

- **Terminal weighting matrix computation.** Equation (5.5) is relaxed $\forall x \in \hat{\mathcal{X}}$ by

$$\|x-x^r\|_{\hat{Q}}^2 \geq \|f(x, \hat{K}(x-x^r)+u^r)-x^r\|_{\hat{Q}}^2+\|x-x^r\|_Q^2+\|\hat{K}(x-x^r)\|_R^2-\beta, \quad (5.18)$$

  where $\hat{Q} \succ 0$ is computed for some $\beta \geq 0$. Various methods exist to compute such $\hat{Q}$, see e.g. [69].

- **Predicted trajectory.** To guarantee recursive feasibility at each iteration of the scheme, the feasible predicted trajectory now needs to be computed as part of a feasible tube $\mathbf{X}^\circ = \{X_k^\circ\}_{k=0}^N$ to enforce constraint satisfaction under all realisations of the additive uncertainty. The feasible tube can be parameterised by means of elementwise bounds as follows

$$X_k^\circ = \{x \,|\, \underline{x}_k^\circ \leq x \leq \overline{x}_k^\circ\}, \, \forall k \in \{0, \dots, N\},$$

  and, initialising with $\underline{x}_0 = \overline{x}_0 = x[n]$, the components of the tube bounds $\underline{x}_k^\circ, \overline{x}_k^\circ$ can be obtained[5] $\forall k \in \{0, \dots, N-1\}$

$$\underline{x}_{k+1}^\circ = \min_{x \in X_k^\circ} \{f_1(x, K_k x + c_k^\circ) - f_2(x, K_k x + c_k^\circ)\} + \underline{w}, \quad (5.19)$$

$$\overline{x}_{k+1}^\circ = \max_{x \in X_k^\circ} \{f_1(x, K_k x + c_k^\circ) - f_2(x, K_k x + c_k^\circ\} + \overline{w}, \quad (5.20)$$

  where $\underline{w} = \min_{w \in \mathcal{W}} w$, $\overline{w} = \max_{w \in \mathcal{W}} w$, and c_k° is defined as the nominal feedforward control such that $u^\circ = K_k x_k^\circ + c_k^\circ$. Note that the optimisation operations in the inequalities above are intended componentwise. To make these operations computationally tractable[6], we have to linearise f_1 in (5.19)

[5] Note that K_k could be computed solving a LQR problem or a DP recursion for the system linearised around a nominal trajectory derived, e.g. by updating the dynamics with the control sequence computed at the previous iteration assuming no disturbance.

[6] We can then easily maximise a convex function and minimise a concave function over a polytopic set by performing a discrete search over the vertices as explained in Section 5.4.

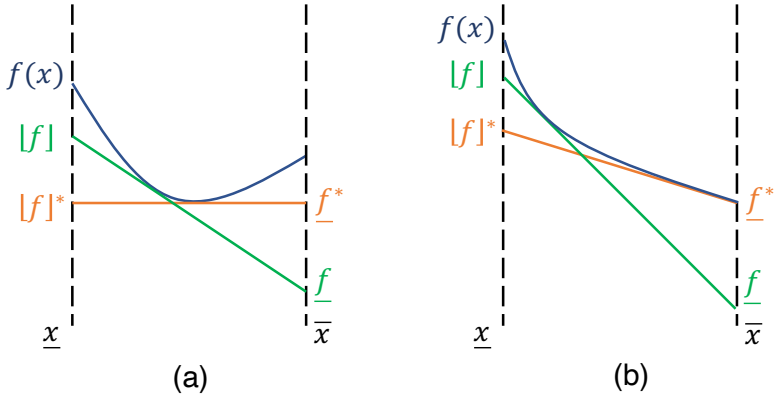


Figure 5.6: First order approximators of a convex function f on a set. (a) The "best" linear underestimator $\lfloor f \rfloor^*$ of f is obtained when linearising the function at its minimiser (orange curve). It is best in the sense that it bounds f tightly and yields the highest lower bound on the set: $\underline{f}^* > \underline{f}$ where $\underline{f}$ is the lower bound given by any other first order underestimator $\lfloor f \rfloor$ of f over the set (green curve). (b) This is true even when the minimum does not exist on the set and the best approximator is instead constructed at the infinimum.

and f_2 in (5.20). We want to linearise the system so that we obtain the "best" first order approximators[7] $\lfloor f_1 \rfloor$, $\lfloor f_2 \rfloor$ in the sense of underestimating the convex functions f_1, f_2 as tightly as possible. More rigorously, we want the approximator that yields the highest lower bound for the function. Such an approximator can be constructed by finding a minimiser for the function over the set, see Figure 5.6 for a 1D example illustrating the concept.

The linearisation points can thus be found by solving the following minimisation problems (componentwise minimisation)

$$\mathbf{x}_k^{\circ,1} = \arg\min_{x \in X_k^\circ} f_1(x, K_k x + c_k^\circ), \tag{5.21}$$

$$\mathbf{x}_k^{\circ,2} = \arg\min_{x \in X_k^\circ} f_2(x, K_k x + c_k^\circ), \tag{5.22}$$

where $\mathbf{x}_k^{\circ,1} = \{x_{k,1}^{\circ,1}, \dots x_{k,n_x}^{\circ,1}\}$ is the set of linearisation states obtained after n_x componentwise minimisations of f_1, $\mathbf{x}_k^{\circ,2} = \{x_{k,1}^{\circ,2}, \dots x_{k,n_x}^{\circ,2}\}$ is the set of linearisation states obtained after n_x componentwise

[7] As a reminder, $\lfloor f \rfloor = f(x^\circ) + \nabla f^\top(x^\circ)(x - x^\circ)$ is the first order approximator (or Jacobian linear approximation) of a function f around linearisation point x°.

minimisations of f_2. This in turns generates two sets of n_x linearisation inputs $\mathbf{u}_k^{\circ,1} = \{K_k x_{k,1}^{\circ,1} + c_k^\circ, \dots K_k x_{k,n_x}^{\circ,1} + c_k^\circ\}$ and $\mathbf{u}_k^{\circ,2} = \{K_k x_{k,1}^{\circ,2} + c_k^\circ, \dots K_k x_{k,n_x}^{\circ,2} + c_k^\circ\}$.

We can now reexpress the tube bounds as a set of computationally tractable optimisation problems as follows

$$\underline{x}_{k+1}^\circ = f_1(\mathbf{x}_k^{\circ,1}, \mathbf{u}_k^{\circ,1}) + \min_{\hat{x}\in\mathcal{V}(X_k^\circ)} \{-f_2(\hat{x}, K_k\hat{x} + c_k^\circ)\} + \underline{w}, \tag{5.23}$$

$$\overline{x}_{k+1}^\circ = \max_{\hat{x}\in\mathcal{V}(X_k^\circ)} \{f_1(\hat{x}, K_k\hat{x} + c_k^\circ)\} - f_2(\mathbf{x}_k^{\circ,2}, \mathbf{u}_k^{\circ,2}) + \overline{w}, \tag{5.24}$$

where $f_i(\mathbf{x}_k^{\circ,i}, \mathbf{u}_k^{\circ,i})$ was used as an abuse of notation for $[f_i(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})]_l$, $\forall l \in \{1, \dots, n_x\}$, $\forall i = 1, 2$ and where $\mathcal{V}(X_k^\circ) = \{\underline{x}_k^\circ, \overline{x}_k^\circ\}$, $\forall k \in \{0, \dots, N-1\}$. Finally, we use the linearisation points to update the linearised models as follows, $\forall i = 1, 2$, $\forall k = 0, \dots, N-1$,

$$A_{i,k} = \frac{\partial f_i}{\partial x}(\mathbf{x}_k^{\circ,i}, \mathbf{u}_k^{\circ,i}), \tag{5.25}$$

$$B_{i,k} = \frac{\partial f_i}{\partial u}(\mathbf{x}_k^{\circ,i}, \mathbf{u}_k^{\circ,i}), \tag{5.26}$$

$$\Phi_{i,k} = A_{i,k} + B_{i,k}K_k, \tag{5.27}$$

$$f_{i,k}^\circ = f_i(\mathbf{x}_k^{\circ,i}, \mathbf{u}_k^{\circ,i}). \tag{5.28}$$

We summarise the method to compute these in Algorithm 3.

Remark 4. *The geometrical method we propose to solve problems* (5.19) *and* (5.20) *can be interpreted in terms of decompositions of the original nonconvex problems into successive simpler optimisation problems. For example, to solve the problem*

$$\max_{x\in\mathcal{P}} \{f_1(x) - f_2(x)\},$$

where f_1, f_2 are convex and $\mathcal{P}$ is a polytope, we can first solve the convex problem

$$x^\circ = \arg\min_{x\in\mathcal{P}} f_2(x),$$

then use the solution of the latter to reduce the former problem into

$$\max_{\hat{x}\in\mathcal{V}(\mathcal{P})} \{f_1(\hat{x}) - f_2(x^\circ)\},$$

which is just a pointwise maximum operation over the discrete set of vertices $\mathcal{V}(\mathcal{P})$.

Remark 5. *It is important to stress that Algorithm 3 generates multiple linearisation points, one for every component of* $f_i, \forall i = 1, 2$, *so* $2n_x$ *points.*

Algorithm 3: Feasible tube generation for linearisation in the presence of additive disturbances.

Input: current state $x[n]$ and nominal feedforward sequence $\mathbf{c}^\circ$
Initialisation: Set $\underline{x}_0^\circ = \overline{x}_0^\circ = x[n]$ so that $X_0^\circ \leftarrow \{x[n]\}$
for $k \leftarrow 0$ **to** $N-1$ **do**
  for $l \leftarrow 1$ **to** n_x **do**
    for $i \leftarrow 1$ **to** 2 **do**
      Solve for the linearisation points:
      $x_{k,l}^{\circ,i} \leftarrow \arg\min_{x \in X_k^\circ} [f_i(x, K_k x + c_k^\circ)]_l$
      $u_{k,l}^{\circ,i} \leftarrow K_k x_{k,l}^{\circ,i} + c_k^\circ$
      Compute the system linearisation about each point:
      $[A_{i,k}]_l \leftarrow \frac{\partial f_i}{\partial x}(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})$, $[B_{i,k}]_l \leftarrow \frac{\partial f_i}{\partial u}(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})$,
      $[\Phi_{i,k}]_l \leftarrow [A_{i,k}]_l + [B_{i,k}]_l K_k$, $[f_{i,k}^\circ]_l \leftarrow [f_i(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})]_l$
    end
    Compute the tube bounds:
    $[\underline{x}_{k+1}^\circ]_l \leftarrow [f_1(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})]_l + \min_{\hat{x} \in \mathcal{V}(X_k^\circ)} \{-[f_2(\hat{x}, K_k \hat{x} + c_k^\circ)]_l\} + [\underline{w}]_l$
    $[\overline{x}_{k+1}^\circ]_l \leftarrow \max_{\hat{x} \in \mathcal{V}(X_k^\circ)} \{[f_1(\hat{x}, K_k \hat{x} + c_k^\circ)]_l\} - [f_2(x_{k,l}^{\circ,i}, u_{k,l}^{\circ,i})]_l + [\overline{w}]_l$
  end
  Update the feasible tube:
  $X_k^\circ \leftarrow \{x \,|\, \underline{x}_k^\circ \leq x \leq \overline{x}_k^\circ\}$
  Gather results:
  $\mathbf{x}_\mathbf{k}^{\circ,\mathbf{i}} \leftarrow \{x_{k,1}^{\circ,i}, \dots x_{k,n_x}^{\circ,i}\}$,
  $\mathbf{u}_\mathbf{k}^{\circ,\mathbf{i}} \leftarrow \{u_{k,1}^{\circ,i}, \dots u_{k,n_x}^{\circ,i}\}$.
end

- **Optimisation.** Optimisation problem (5.16) should be modified to account for the external disturbance and the new linearisation scheme in Algorithm 3. Defining the state tube[8] $\mathbf{X} = \{X_k\}_{k=0}^N$ by means of elementwise bounds as follows $X_k = \{x \,|\, \underline{x}_k \leq x \leq \overline{x}_k\}$, $\mathcal{V}(X_k) = \{\underline{x}_k, \overline{x}_k\}$, $\forall k \in \{0, \dots, N\}$ and

[8]Not to be confused with the feasible tube used to generate the linearisation points in Algorithm 3.

initialising this sequence by the (measured) state $x[n]$ at time n, we can formulate the following problem

$$\begin{aligned}
&\min_{\mathbf{c},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\chi}} \ ||\theta_N||^2_{\hat{Q}} + \sum_{k=0}^{N-1} \Big[||\theta_k||^2_Q + ||\chi_k||^2_R \Big] \\
&\text{subject to, } \forall k \in \{0,\dots,N-1\},\ \forall \hat{x}_k \in \mathcal{V}(X_k): \\
&\underline{x}_{k+1} \leq f^{\circ}_{1,k} + \Phi_{1,k}(\hat{x}_k - \mathbf{x}^{\circ,1}_k) + B_{1,k}c_k - f_2(\hat{x}_k, c^{\circ}_k + c_k + K_k\hat{x}_k) + \underline{w} \\
&\overline{x}_{k+1} \geq f_1(\hat{x}_k, c^{\circ}_k + c_k + K_k\hat{x}_k) - (f^{\circ}_{2,k} + \Phi_{2,k}(\hat{x}_k - \mathbf{x}^{\circ,2}_k) + B_{2,k}c_k) + \overline{w} \\
&\hat{x}_k \subseteq \mathcal{X}, \\
&c^{\circ}_k + c_k + K_k\hat{x}_k \subseteq \mathcal{U} \\
&\theta_k \geq |\hat{x}_k - x^r| \\
&\chi_k \geq |c^{\circ}_k + c_k + K_k\hat{x}_k - u^r| \\
&\theta_N \geq |\hat{x}_N - x^r| \\
&\|\hat{x}_N - x^r\|^2_{\hat{Q}} \leq \hat{\gamma}, \forall \hat{x}_N \in \mathcal{V}(X_N) \\
&\underline{x}_0 = \overline{x}_0 = x[n].
\end{aligned} \tag{5.29}$$

where by abuse of notation, we use $\phi_{i,k} = \Phi_{i,k}\mathbf{x}^{\circ,i}_k$ instead of $[\phi_{i,k}]_l = [\Phi_{i,k}]_l x^{\circ,i}_{k,l}$, $\forall l \in \{1,\dots,n_x\}$, $\forall i = 1,2$, and $|x_k|$ instead of $\big|[x_k]_l\big|$, $\forall x_k \in \mathbb{R}^{n_x}$, $\forall l \in \{1,\dots,n_x\}$.

- **Update.** Initialising the algorithm with $\mathbf{c} \leftarrow \{u^r - \hat{K}x^r, \dots, u^r - \hat{K}x^r\}$, the solution of (5.29) for $\mathbf{c}$ is used at each iteration k to update $\mathbf{c}^{\circ}$ by setting

$$c^{\circ}_k \leftarrow c^{\circ}_k + c_k,$$

for $k = 0,\dots N-1$. At each time step n, the following controller is implemented

$$u[n] = K_0 x[n] + c^{\circ}_0,$$

and we update $\mathbf{c}^{\circ}$ with the tail sequence

$$\mathbf{c}^{\circ} \leftarrow \{c^{\circ}_1, \dots, c^{\circ}_{N-1}, u^r - \hat{K}x^r\}.$$

We summarise all steps in Algorithm 4. An extension of the theorems in Section 5.5 to that new algorithm is presented below.

Algorithm 4: DC-TMPC algorithm with additive disturbances.

Initialisation
  Compute $\hat{K}$, $\hat{Q}$, $\hat{\gamma}$ satisfying (5.18), (5.6) and (5.7)
  Compute initial nominal feedforward $\mathbf{c}^\circ$ such that Algorithm 3 generates a feasible tube $\mathbf{X}^\circ$.
end
Online at times $n = 0, 1, \ldots$
  Obtain $x[n]$ and set $\|\mathbf{c}\| \leftarrow \infty$, $j \leftarrow 0$
  while $\|\mathbf{c}\| >$ *tolerance* & $j <$ *maxIters* **do**
    Compute K_k, $\forall k = 0, \ldots, N-1$, using (5.8) for the undisturbed linear system.
    Compute $\mathbf{x}^{\circ,i}, \mathbf{u}^{\circ,i}$, $A_{i,k}$, $B_{i,k}$, $\Phi_{i,k}$, $f^\circ_{i,k}$, $\forall i = 1, 2$, $\forall k = 0, \ldots, N-1$, using Algorithm 3.
    Solve (5.29) to find the optimal $\mathbf{c}$ and $\mathbf{X}$
    Update $\mathbf{c}^\circ \leftarrow \mathbf{c}^\circ + \mathbf{c}$,
    Update the iteration counter: $j \leftarrow j + 1$
  end
  Update the control input: $u[n] \leftarrow K_0 x[n] + c^\circ_0$
  Update $\mathbf{c}^\circ \leftarrow \{c^\circ_1, \ldots, c^\circ_{N-1}, u^r - \hat{K}x^r\}$
end

Proposition 7 (Recursive feasibility with additive disturbance)**.** *Assuming initial feasibility at time $n = 0$ and iteration $j = 0$, problem (5.29) in Algorithm 4 is feasible at all times $n > 0$ and all iterations $j > 0$.*

Proof. See [135] for a proof. □

Theorem 8 (Stability with additive disturbance)**.** *The control law of Algorithm 4 ensures that the system 5.17 satisfies the state and input constraints and the following quadratic stability condition on the average stage cost:*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{n=0}^{t-1} ||x[n] - x^r||_Q^2 + ||u[n] - u^r||_R^2 \leq \beta,$$

where β satisfies equation (5.18)*.*

Proof. From 5.18 the cost satisfies

$$J^{j_{n+1},n+1} \leq J^{j_n,n} - \|x[n] - x^r\|_Q^2 - \|u[n] - u^r\|_R^2 + \beta,$$

where $J^{j_n,n}$ is the cost at final iteration for time step n. This yields after summation, and by the telescopic sum

$$\frac{1}{t}(J^{j_{t-1},t-1} - J^{j_0,0}) + \frac{1}{t}\sum_{n=0}^{t-1}(\|x[n] - x^r\|_Q^2 + \|u[n] - u^r\|_R^2) \leq \beta.$$

Finally taking the limit as $t \to \infty$ yields

$$\lim_{t\to\infty} \frac{1}{t}\sum_{n=0}^{t-1}(\|x[n] - x^r\|_Q^2 + \|u[n] - u^r\|_R^2) \leq \beta,$$

which guarantees a finite bound on the averaged stage cost. □

It follows from Theorem 8 that the objective of the MPC optimisation is finite because the cost function has a finite number of stages and each stage cost is finite.

5.7.2 Twice-continuously-differentiable dynamical systems

The DC-TMPC algorithm can be further generalised to systems whose dynamics are twice continuously differentiable

$$x[n+1] = f(x[n], u[n]), \quad f \in \mathcal{C}^2.$$

This is due to a result known since Hartman [127]: any twice-continuously-differentiable function can be expressed in DC form.

To the best of our knowledge, there is no systematic way to obtain an exact DC representation. However, we describe below several methods to obtain an approximate DC representation.

Theorems of Section 5.5 apply as long as the nonlinear dynamics can be approximated arbitrarily closely by the approximation model. Approximation errors can be estimated and treated as bounded additive disturbances[9] as in Section 5.7.1.

[9]This supposes that a bound on the approximation error can be computed. This can either be estimated empirically by offline evaluation of the maximum error between the function and approximation model. Alternatively, an upper bound can be computed using theoretical results. For example, in the case of approximations by means of shallow neural networks, a bound can be computed that depends on the number of nodes, function input dimension and number of training samples, see [136]. Once a worst case bound has been computed, it can be treated as an external disturbance in the TMPC framework.

Polynomial approximation

As discussed already in Chapter 4, a DC decomposition can be obtained using polynomial approximation. We have presented a technique that applies to functions of one variable. We present here a more general method that applies to a multivariate scalar function to compute the DC decomposition $f = g - h$, where g, h are convex. This method applies to a large class of systems since: a) any continuous function can be approximated arbitrarily closely by a polynomial (Stone-Weierstrass theorem) and; b) any polynomial can be computed as a difference of convex polynomials of same degree [131].

1) *Fit polynomial to data.* Assume that the nonlinear model[10] f can be approximated arbitrarily closely by a polynomial of degree $2d$ in Gram form such that $f \approx y(x)^\top F y(x)$, where $F = F^\top$ is the Gram matrix and $y = [1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d]^\top$ is a vector of monomials of degree up to d (y has size $C^{|x|}_{d+|x|}$). Generate N_s samples $f_s = f(x_s), \forall s \in [1, ..., N_s]$ of the nonlinear model and solve the following least squares problem:

$$\mathcal{LS}: \quad \min_F \quad \sum_{s=0}^{N_s} \|f_s - y(x_s)^\top F y(x_s)\|_2^2, \quad \text{s.t.} \quad F = F^\top.$$

2) *Compute the Hessians of the decomposition.* Let $g(x) \approx y(x)^\top G y(x)$ and $h(x) \approx y(x)^\top H y(x)$ be convex polynomials such that their Hessians $d^2g(x)/dx^2 = y(x)^\top H_g y(x)$ and $d^2h(x)/dx^2 = y(x)^\top H_h y(x)$ are positive semidefinite (PSD). Finding G, H such that $F = G - H$ and $h(x)$, $g(x)$ are convex reduces to solving the following semidefinite program (SDP)

$$\mathcal{SDP}: \max_{G,\sigma} \quad \sigma \quad \text{s.t.} \quad H_g(G) \succeq \sigma I, \quad H_h(G - F) \succeq \sigma I,$$

with $\forall i, j$

$$[H_g]_{ij} = D_{j,i}^\top G + G D_{i,j} + D_i^\top G D_j + D_j^\top G D_i,$$
$$[H_h]_{ij} = D_{j,i}^\top (G - F) + (G - F) D_{i,j} + D_i^\top (G - F) D_j + D_j^\top (G - F) D_i,$$

where I is the identity matrix of compatible dimensions, D_i is a matrix of coefficients such that $dy/dx_i = D_i y$ and $D_{i,j} = D_i D_j$.

[10] Note that it does not need to be a mathematical function but can be defined from data.

Input-convex neural networks

A neural network architecture with convex input-output map was introduced in [137]. The so called input-convex neural network (ICNN) is obtained by constraining the kernel weights of the network to be nonnegative at training stage and using convex activation functions. A fully connected L-layer feedforward ICNN with parameters $\theta = \{\Theta_{1:L-1}, \Phi_{0:L-1}, b_{0:L-1}\}$ and input-output map given by $z_L = f(x;\theta)$ can be defined by the following update equation $\forall l \in \{0, \dots, L-1\}$ (cf. Figure 5.7)

$$z_{l+1} = \sigma(\Theta_l z_l + \Phi_l x + b_l),$$

where z_l is the layer activation (with $z_0, \Theta_0 = 0$), Θ_l are positively constrained kernel weights ($\{\Theta_l\}_{ij} \geq 0$, $\forall i, j$, $\forall l \in \{1, \dots, L-1\}$), Φ_l are input weights, b_l are bias weights, and $\sigma(\cdot)$ is a convex activation function (e.g. the ReLU function).

Each layer of an ICNN thus consists in the composition of a convex function with a nondecreasing convex function, which results in a convex map f between the input of the network and the output (see Appendix A, Remark 11 or [138]). When the input and output are taken as $x = (x_k, u_k)$ and $z_L = x_{k+1}$, the network can be used to learn the dynamic function of a system. Note that in order to allow sufficient representation power, the input is fed-back to each layer.

Recently, a neural network model with DC structure was proposed in [139], and consists in two parallel ICNN whose outputs are subtracted. They also note that a similar DC structure can be achieved with the same architecture as presented in Figure 5.7 where the kernel weights Θ_{L-1} connecting the two last layers are unconstrained. Such a network (which we refer to as DCNN in the sequel) can be trained on data collected from "smooth" dynamical systems to learn the dynamics in DC form.

To illustrate the concept with a 1D example, we consider the problem of learning the sine function as a difference of convex functions. A DCNN with 3 hidden layers, 64 units and ReLU activation was trained on 100,000 input-output samples generated randomly from the function $f = \sin(x)$ using the open-source neural-network library Keras with the RMSprop optimiser, mean squared error (MSE) loss, batch size of

64, 10 epochs [140]. A test accuracy of 0.02 was recorded using a mean absolute error (MAE) metrics on 100 test samples. Figure 5.8 shows the decomposition obtained when the network learns the sine function. The prediction by the network (plain red) matches with the actual function (dashed blue) and is obtained as the difference of two convex functions learned by the network (plain blue and green).

To demonstrate its capabilities, the DCNN architecture was leveraged in the present case study of the coupled tank model. The network has 1 hidden layer, 64 units and ReLU activation and was trained on 100,000 input-output samples generated randomly from the tank model with the RMSprop optimiser, MSE loss, batch size of 64, 10 epochs. The evolution of the training loss and validation loss (obtained using a validation split in the training data of 20%) as a function of training epochs are recorded in Figure 5.9 to ensure that the model does not overfit the training data. An accuracy of 0.036 was obtained on 100 test samples with a MAE metrics. The results of the approximation of the dynamics function of the coupled tank by a DCNN are presented in Figure 5.10 for a given input voltage of 12 V. The nonconvex surfaces defining the dynamics update for a given voltage (blue) are approximated by the DCNN as a difference of two convex surfaces (green and orange dots).

A major advantage of techniques based on neural networks is that they can be leveraged very efficiently in the context of tube-based MPC with the DC-TMPC algorithm. Indeed, they allow one to express the tube constraint in terms of affine transformations $\mathcal{A}_{\theta_l,x}(z) = \Theta_l z + \Phi_l x + b_l$, $\forall l \in \{1, \ldots, L-1\}$ and $\mathcal{A}_{\Phi_0}(x) = \Phi_0 x + b_0$, and pointwise maximum operations $\sigma(x) = \max\{x, 0\}$ (with ReLU activation) which are potentially more efficient than polynomial constraints:

$$f(x;\theta) = (\sigma \circ \mathcal{A}_{\theta_{L-1},x} \circ \ldots \circ \sigma \circ \mathcal{A}_{\theta_1,x} \circ \sigma \circ \mathcal{A}_{\Phi_0})(x),$$

Moreover, the backpropagation algorithm is insightful in determining the first order derivative of the network in terms of the chain rule for the successive linearisation steps in the DC-TMPC paradigm:

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x;\theta) = (H \circ \Theta_{L-1} \circ \ldots \circ H \circ \Theta_1 \circ H \circ \Phi_0)(x),$$

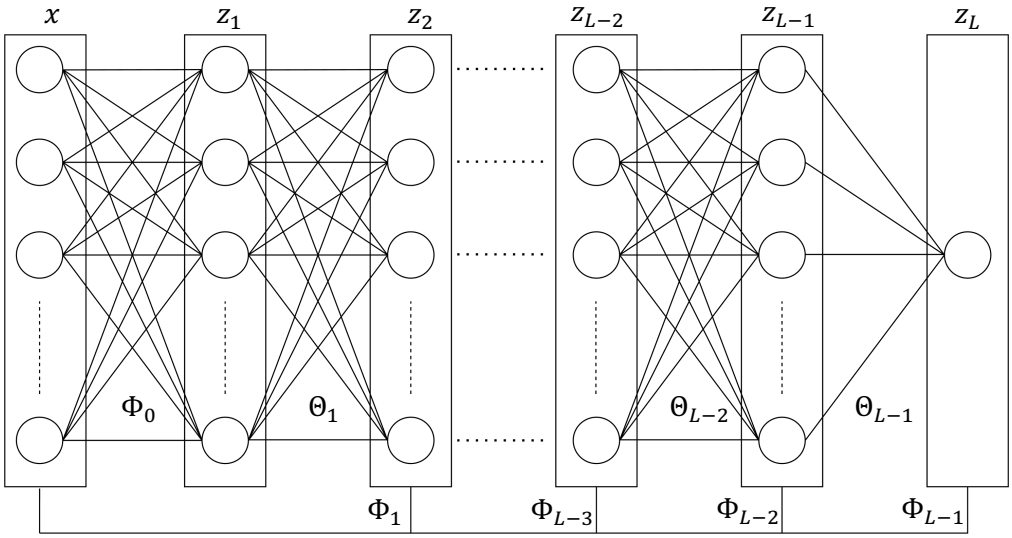


Figure 5.7: ICNN architecture $z_L = f(x;\theta)$ where the kernel weights Θ_l are nonnegative $\forall l \geq 1$ and activation functions are convex.

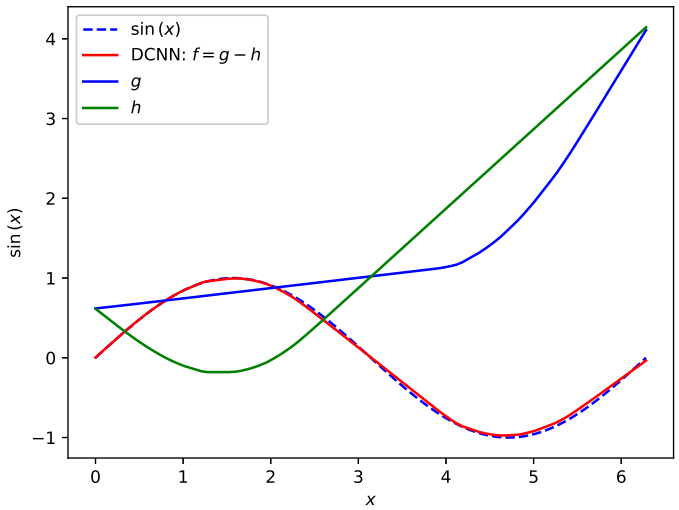


Figure 5.8: Approximation of $f = \sin(x)$ by a DCNN.

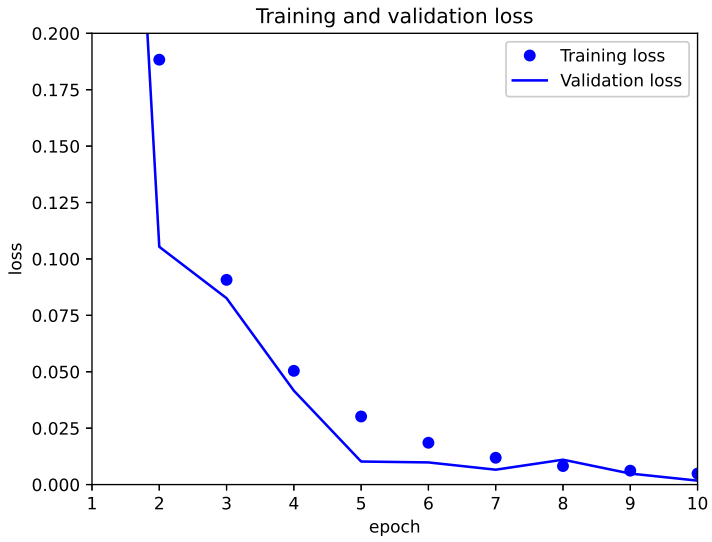


Figure 5.9: Evolution of the training and validation losses over the training process to learn the coupled tank model with a DCNN.

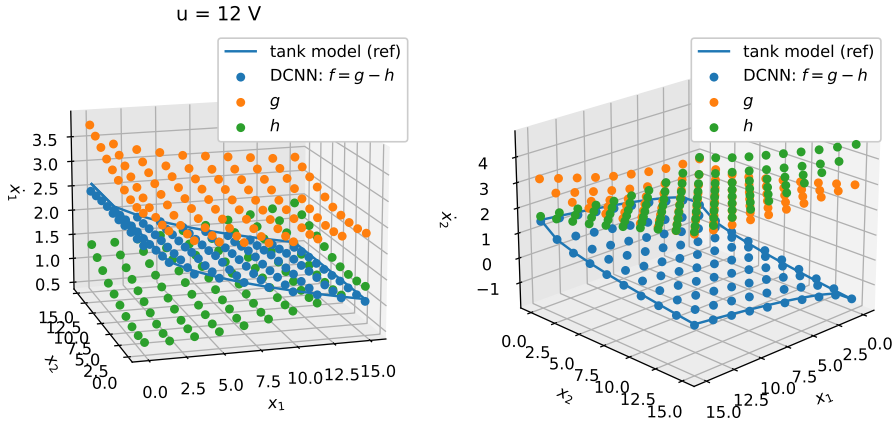


Figure 5.10: Approximation of the tank model as a difference of convex functions by a DCNN.

where $H(\cdot)$ is the Heaviside step function.

Future work will investigate the use of neural networks in that context and will compare them with other DC decomposition techniques.

Recurrent neural network approximation

Similarly to feedforward neural networks, it is possible to construct recurrent neural networks (RNN) with a special DC structure [139, 141]. This time they are used to approximate the entire time series or trajectory $\mathbf{x} = \{x_k\}_{k=1}^N$ where N is the horizon for a given input sequence $\mathbf{u} = \{u_k\}_{k=1}^N$ and initial condition x_0. A difference of convex RNN (DCRNN) neural network defining a mapping $\mathbf{x} = f(\mathbf{u}; x_0, \theta)$ can be described by the following update equations $\forall l \in \{1, \ldots, N\}$

$$z_l = \sigma(A_{l-1} z_{l-1} + B_l u_l + b_l), \quad z_0 = x_0 \tag{5.30}$$

$$x_l = C_l z_l, \tag{5.31}$$

where A_{l-1} are the positively constrained kernel weights ($\{A_l\}_{ij} \geq 0$, $\forall i, j$, $\forall l \in \{0, \ldots, N-1\}$), B_l the input weights, C_l the output weights, b_l the bias, and $\sigma(\cdot)$ is a convex activation function (e.g. ReLU). Note that since the C_l are unconstrained, the network defines a DC input-output map.

We have trained a DCRNN network implemented in Keras over a horizon $N = 50$, 1 hidden layer, 8 units and ReLU activation on 50,000 input-output training sequences generated by sampling the coupled tank model. We used the RMSprop optimiser, MSE loss and trained by batch of 64 over 20 epochs. A MAE testing accuracy of 2.45 is recorded, and a typical trajectory prediction for a random initial condition and random input sequence is shown in Figure 5.11. We note that the fit could be potentially improved by varying the hyperparameters of the network (e.g. by reducing the horizon N, adding extra hidden layers) and generating more training data. The latter might be impractical depending on the application. Also the predictions are much accurate for x_1 than x_2 (probably because the error to estimate the height in the tank upstream propagates to the tank downstream), so two different networks could be used with different sets of hyperparameters (hidden units, hidden layers, activation) tailored specifically for each state.

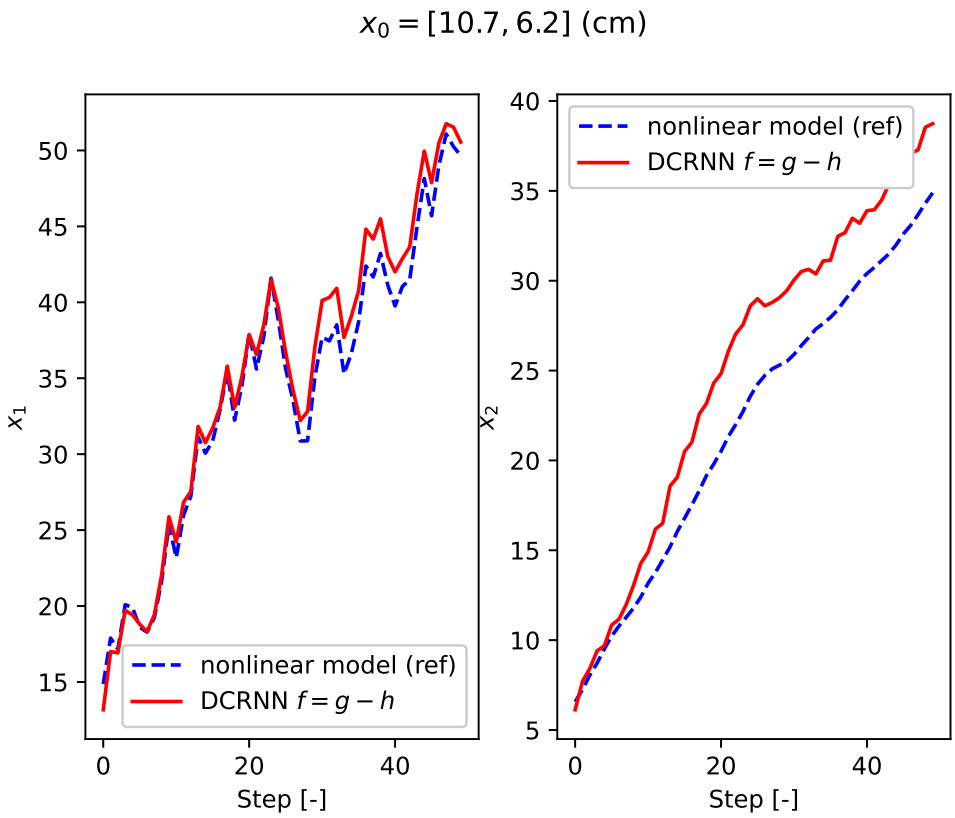


Figure 5.11: Prediction of the coupled tank trajectories by a DCRNN for given random initial condition and input sequence.

Radial basis functions approximation

A radial basis function (RBF) is a function φ whose value depends only on the distance between its input x and some fixed point c (or centre) , i.e $\varphi(r) = \varphi(||x - c||)$ where $r = ||x - c||$.

A powerful approach in machine learning is to approximate functions using weighted sums of RBF as follows

$$f = \sum_{j=0}^{m} \alpha_j \varphi(||x - c_j||_{\rho_j}),$$

where $\alpha_j \in \mathbb{R}$ are coefficients, $\rho_j \in \mathbb{S}_+^n$ are scaling matrices, $c_j \in \mathbb{R}^n$ are centres and φ is RBF.

If the RBF is a convex function, then $f = g - h$ is DC where

$$g = \sum_{j=0}^{m} \max\{\alpha_j, 0\}\varphi(||x - c_j||_{\rho_j}), \quad h = -\sum_{j=0}^{m} \min\{\alpha_j, 0\}\varphi(||x - c_j||_{\rho_j}).$$

The following theorem gives sufficient conditions for a RBF to be convex.

Theorem 9 (Convex RBF)**.** *A RBF $\varphi(r(x))$ where $r(x) = ||x - c||$ is convex with respect to x if $\varphi'(r) \geq 0$ and $\varphi''(r) \geq 0$.*

Proof. By the second order convexity condition (Appendix A, Theorem 11), the RBF φ is convex iff the Hessian is PSD, i.e. $\nabla^2\varphi(x) \succeq 0$, $\forall x \in \mathbb{R}^n$.

By the chain rule, the gradient is given by

$$\nabla\varphi = \varphi'(r)\nabla r,$$

and the Hessian is

$$\nabla^2\varphi = \varphi''(r)\nabla r(\nabla r)^\top + \varphi'(r)\nabla^2 r.$$

The Hessian is PSD iff, $\forall z \in \mathbb{R}^n$

$$\begin{aligned} z^\top\nabla^2\varphi z \geq 0 &\Leftrightarrow \varphi''(r)z^\top\nabla r(\nabla r)^\top z + \varphi'(r)z^\top\nabla^2 r z \geq 0, \\ &\Leftrightarrow \varphi''(r)||(\nabla r)^\top z||_2^2 + \varphi'(r)z^\top\nabla^2 r z \geq 0, \end{aligned} \tag{5.32}$$

By convexity of the norm (Appendix A, Remark 10), $z^\top\nabla^2 r z \geq 0$, and since $||(\nabla r)^\top z||_2^2 \geq 0$, the Hessian is PSD if $\varphi''(r) \geq 0$, $\varphi'(r) \geq 0$. □

Remark 6. *An example of a convex RBF is the multiquadric function*

$$\varphi(r) = \sqrt{1 + (\varepsilon r)^2},$$

where ε is a parameter.

Remark 7. *There are several approaches to learning the parameters c_j, ρ_j, α_j, $\forall j = 1, \ldots, m$. Given c_j and ρ_j, the weights α_j can be obtained simply by solving a least squares problem. If all parameters are to be optimised together, algorithms similar to training of neural networks can be used [142].*

5.8 Conclusions

This chapter introduces DC-TMPC, a new method for robust nonlinear MPC applied to systems representable as a difference of convex functions. The method relies on successively approximating the system dynamics around predicted trajectories and exploits convexity in the linearisation errors to construct robust and non-conservative tubes containing the perturbed trajectories. Convergence, recursive feasibility and asymptotic stability of the proposed algorithm are demonstrated. The algorithm is applied to regulation of fluid levels in a coupled tank system. Finally we have discussed extension of the method to i) systems subject to additive disturbance; ii) a broader class of dynamical systems whose dynamics present sufficient smoothness properties (namely $\mathcal{C}^2$ differentiability).

One major drawback of the method is the exponential explosion of the number of constraints with the number of states. In practice, this is mitigated by the small number of states of many practical systems (for example in the last chapter, only two states were needed to define the uncertainty sets for the trajectory optimisation problem of a VTOL aircraft). We note also that this number can be reduced using prior knowledge of the coefficients of the terms which appear linearly in the states in the inequalities. Further reductions are possible with a parameterisation of the tube in terms of low complexity polytopes (e.g. simplex uncertainty sets in which the number of vertices scales linearly with the number of states), but such an approach would be more conservative. Moreover, since the algorithm can be terminated after any number of iterations (e.g. after a single solver iteration) without affecting stability, feasibility or constraint satisfaction, the computation time can be further reduced (but this comes at the expense of accuracy).

Future work could include demonstrating that the method converges to a point satisfying first order optimality conditions, using other tube parameterisations in order to reduce the number of constraints (e.g. simplexes), and investigating applications to problems such as robust trajectory generation for VTOL aircraft with systematic methods of computing DC decompositions. The latter extension is the object of the next chapter.

6

Robust MPC of VTOL aircraft

Contents

6.1 Introduction

We revisit the urban air mobility (UAM) scenario introduced in Chapter 4 to achieve robust control during the transition phase of tiltwing vertical take-off and landing (VTOL) aircraft in the presence of wind disturbances, leveraging the tube-based model predictive control (TMPC) paradigm developed in Chapter 5. In the

latter, we introduced the DC-TMPC algorithm for systems whose dynamics can be formulated as a difference of convex (DC) functions. The main idea is to successively linearise the dynamics around predicted trajectories and treat linearisation errors as bounded disturbances. Because the linearised functions are convex, so are their linearisation errors, and since these errors are maximised at the boundary of the domain on which they are evaluated, they can therefore be bounded tightly. The trajectories of model states are thus bounded tightly by a sequence of sets (or tube) defined by convex inequalities that form the basis of a multi-stage optimal and robust controller. We showed that the DC-TMPC algorithm offers guarantees of convergence, robust stability and recursive feasibility. Although very efficient, the scope of applicability is initially limited to systems with DC dynamics. Moreover we raised concerns about the scalability of the method due to the exponential increase of constraints with the number of states. In this chapter, we want to address these two drawbacks and show that: i) the applicability of the algorithm can be considerably extended if DC decompositions techniques are used (see Chapter 5, Section 5.7.2); ii) the algorithm can be applied to solve real control engineering problems, namely the transition of a VTOL aircraft in the presence of wind gusts.

This chapter is organised as follows. We start by developing a mathematical model of a tiltwing VTOL aircraft subject to wind disturbance in Section 6.2. In Section 6.3, we formulate the MPC optimisation problem and leverage a DC decomposition of the nonlinear dynamics to construct robust tubes for the state trajectories. Section 6.4 discusses simulation results obtained for a case study based on the Airbus A^3 Vahana. Section 6.5 presents conclusions.

6.2 Modeling

We consider again the transition of tiltwing VTOL aircraft as in Chapter 4. However, we now tackle the (robust) control problem for tiltwing VTOL aircraft and we introduce here additional complexities to the model: wind disturbance, data-based aerodynamic modeling of the wing, reparameterisation of the equations of motion

(EOM) in cartesian coordinates for better numerical stability. For convenience we thus restate the EOM here to ease the discussion.

6.2.1 Assumptions

We assume a flat earth model and consider a longitudinal point-mass model. Tiltwing aircraft are considered (see Figure 6.1), with one or more aerodynamic surfaces carrying thrust effectors that can be rotated by an actuator through a range of 90 degrees as the aircraft transitions between wing-borne and thrust-borne flight. We consider a scenario where the aircraft is subject to wind disturbances (modeled as additive uncertainty). We further assume classical inner/outer loop flight control laws for stabilisation of aircraft attitude with time-scale separation, so that the closed loop pitch dynamics are much faster than those of the desired flight path and tiltwing actuation. Consequently the pitch angle θ (defined here as the angle of the fuselage axis from horizontal earth plane) is assumed to be maintained at all times by the attitude control loop at a constant reference angle $\theta^r = 0$.

6.2.2 Equations of motion

The EOM with respect to inertial frame O_{XZ} are given by[1]

$$\dot{X} = V_x, \quad \dot{Z} = V_z, \tag{6.1}$$

$$m\dot{V}_x = \underbrace{T\cos(\alpha+\gamma) - D\cos\gamma - L\sin\gamma}_{f_1} + W_x, \tag{6.2}$$

$$m\dot{V}_z = \underbrace{-T\sin(\alpha+\gamma) + D\sin\gamma - L\cos\gamma + mg}_{f_2} + W_z, \tag{6.3}$$

where T is the thrust magnitude, L, D are the lift and drag forces, V_x, V_z the components of the aircraft velocity in the inertial frame $V = \sqrt{V_x^2 + V_z^2}$, α is the angle of attack, $\gamma = -\arcsin(V_z/V)$ is the flight path angle (defined as the angle of the velocity vector from horizontal), and (X, Z) the position in inertial frame.

[1]Note that these equations are similar to equations (4.1)-(4.4) in Chapter 4 but we introduce here additive disturbances to the dynamics and express the EOM in cartesian coordinates for better numerical conditioning (the polar form introduced a singularity around hover, $V = 0$).

Wind gusts are modeled by additive bounded disturbances W_x and W_z, which are assumed to lie at all times within known bounds.

The dynamics of the rotating wing are given by

$$J_w\dot{\zeta} = M, \quad \dot{i}_w = \zeta, \tag{6.4}$$

where J_w is the rotational inertia of the wing (about the y-axis), M is the total torque delivered by the tilting actuators, ζ is the tiltwing rate and i_w is the tiltwing angle. The angles i_w, θ, α and γ are related by (see Figure 6.1)

$$i_w + \theta = \alpha + \gamma. \tag{6.5}$$

The effective (blown) velocity V_e and effective (blown) angle of attack α_e seen by the wing due to the effect of the propeller wake on the wing are given by

$$\alpha_e = \arcsin\left(\frac{V}{V_e}\sin\alpha\right), \tag{6.6}$$

$$V_e = \sqrt{V^2 + \frac{2T}{\rho A n}}, \tag{6.7}$$

where ρ is the air density, A the rotor disk area and n the number of propellers. The total lift and drag are modeled as the weighted sum of the blown and unblown terms as follows[2]

$$L = \tfrac{1}{2}\lambda\rho S C_L(\alpha_e)V_e^2 + \tfrac{1}{2}(1-\lambda)\rho S C_L(\alpha)V^2, \tag{6.8}$$

$$D = \tfrac{1}{2}\lambda\rho S C_D(\alpha_e)V_e^2 + \tfrac{1}{2}(1-\lambda)\rho S C_D(\alpha)V^2, \tag{6.9}$$

where S is the wing area, λ is a weighting term representing the portion of the wing in the wake, and C_L and C_D are lift and drag coefficients. We define C_L and C_D using data derived from the Tangler-Ostowari post-stall model [62] (see Figure 6.2), which determines lift and drag forces over a wide range of α and α_e values as the wing operates at high angles of attack during transitions.

[2]This is different from the lift and drag model in equations (4.11)-(4.12) from Chapter 4 where the problem separation trick required to assume, restrictively, that the wing was fully immersed in the flow. We relax that assumption and assume that portions of the wing are unaffected by the propeller wake.

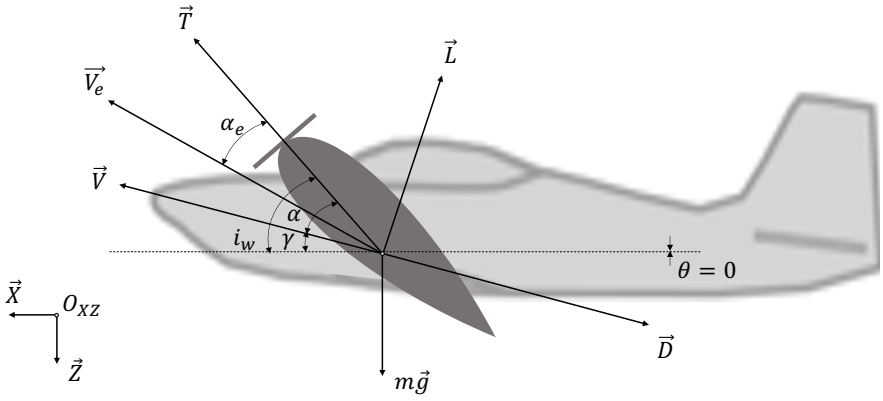


Figure 6.1: Force and velocity definitions for a VTOL aircraft.

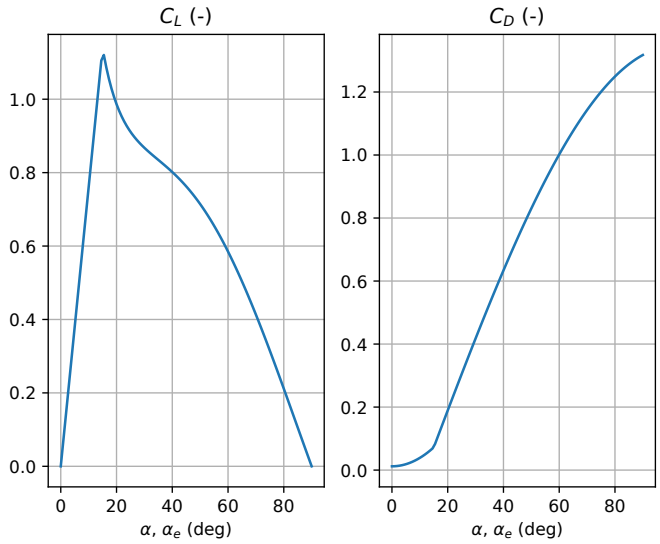


Figure 6.2: Lift and drag coefficients as a function of angle of attack.

6.2.3 Constraints

For the problem considered in this thesis we assume that the gust disturbances W_x, W_z in the forward and vertical directions are bounded: $W_i \in [\underline{W_i}, \overline{W_i}]$ for $i = x, z$. Since wing tilting actuators have finite torque capacity we also assume bounded wing acceleration/deceleration rates given by $\underline{M}$ and $\overline{M}$. To ensure sensible trajectories for passenger comfort and g-loads, we further introduce constraint limits on horizontal ($\dot{V}_x$) and vertical ($\dot{V}_z$) accelerations. Finally, constraints on thrust

range, tilt angle range and absolute velocities can all be expressed in the compact form as input and state constraints,

$$\underline{V}_x \leq V_x \leq \overline{V}_x, \quad \underline{V}_z \leq V_z \leq \overline{V}_z, \quad 0 \leq T \leq \overline{T}, \quad \underline{i}_w \leq i_w \leq \bar{i}_w, \tag{6.10}$$

$$V_x(t_0) = V_{x,0}, \quad V_z(t_0) = V_{z,0}, \quad i_w(t_0) = i_0, \tag{6.11}$$

$$\underline{W}_x \leq W_x \leq \overline{W}_x, \quad \underline{W}_z \leq W_z \leq \overline{W}_z, \tag{6.12}$$

$$\underline{a} \leq \dot{V}_x \leq \overline{a}, \quad \underline{a} \leq \dot{V}_z \leq \overline{a}, \tag{6.13}$$

$$\underline{M}/J_w \leq \ddot{i}_w \leq \overline{M}/J_w. \tag{6.14}$$

6.3 Robust MPC formulation

We now introduce a robust predictive control law for the system presented in Section 6.2. Assuming we are only concerned with velocity control[3], the states (X, Z) can be computed *a posteriori* via (6.1) and thus eliminated from the analysis. Moreover, combining equations (6.2)-(6.9), we can further eliminate the flight path angle, angle of attack, and tiltwing rate from the formulation and express the dynamics with only two states V_x, V_z and two inputs i_w, T as follows

$$m\dot{V}_x = f_1(V_x, V_z, i_w, T) + W_x, \tag{6.15}$$

$$m\dot{V}_z = f_2(V_x, V_z, i_w, T) + W_z, \tag{6.16}$$

where i_w is now an input constrained by its second order derivative (see equation 6.14).

To setup the MPC problem we use the trajectory constraints given by (6.10) - (6.14) together with the terminal set for the velocities

$$\left\| \begin{bmatrix} V_x(t_f) - V_x^r(t_f) \\ V_z(t_f) - V_z^r(t_f) \end{bmatrix} \right\|_{\hat{Q}}^2 \leq \hat{\gamma}, \tag{6.17}$$

where the notation $\cdot^r$ was used to denote a reference to be tracked, t_f is a fixed terminal time, $\hat{\gamma}$ and $\hat{Q} \succ 0$ are respectively a terminal set bound and

[3]This is sensible under the assumption that the control architecture is decoupled between attitude and velocity dynamics as is often the case in aerospace applications thanks to dynamics time-scale separation.

penalty matrix that can be chosen to satisfy equation (5.18) from Chapter 5. Note that equation (6.17) enforces the conditions: $|V_x(t_f) - V_x^r(t_f)| \leq \delta^V$, $|V_z(t_f) - V_z^r(t_f)| \leq \delta^V$, $|i_w(t_f) - i_w^r(t_f)| \leq \delta^{i_w}$, $|T(t_f) - T^r(t_f)| \leq \delta^T$, where $\delta^V, \delta^{i_w}, \delta^T$, are terminal sets bounds.

The control objective is to achieve tracking of a reference trajectory V_x^r, V_z^r, i_w^r, T^r while rejecting the wind disturbances W_x, W_z acting on the system. To do so, at each time step, we could compute a receding horizon control law that minimises the worst case quadratic objective defined for $Q \succ 0$, $R \succ 0$ by

$$J(u) = \max_{\substack{W_x \in [\underline{W}_x, \overline{W}_x] \\ W_z \in [\underline{W}_z, \overline{W}_z]}} \left\{ \left\| \begin{bmatrix} V_x(i_w(t_f), T(t_f), W_x(t_f), W_z(t_f)) - V_x^r(t_f) \\ V_z(i_w(t_f), T(t_f), W_x(t_f), W_z(t_f)) - V_z^r(t_f) \end{bmatrix} \right\|_{\hat{Q}}^2 \right.$$
$$\left. + \int_{t_0}^{t_f} \left\| \begin{bmatrix} V_x(i_w(t), T(t), W_x(t), W_z(t)) - V_x^r(t) \\ V_z(i_w(t), T(t), W_x(t), W_z(t)) - V_z^r(t) \end{bmatrix} \right\|_Q^2 + \left\| \begin{bmatrix} i_w(t) - i_w^r(t) \\ T(t) - T^r(t) \end{bmatrix} \right\|_R^2 \mathrm{d}t \right\} \tag{6.18}$$

subject to (6.10)-(6.17). Note that the states are uncertain and that the state trajectories in the objective are computed as the worst case realisation of the time-varying gust disturbances W_x, W_z. We would then apply the first element of the obtained optimal control sequence, update the current state and input and repeat the process at each time step. Since the model includes the nonlinear functions f_1 and f_2 and that the state is uncertain, computing the control law would require solving a min-max nonlinear program (NLP) which is intractable in practice. In order to leverage the tools developed in Chapter 5 and obtain a computationally efficient implementation of the robust MPC problem, we introduce a DC decomposition of these nonlinear functions. Using the procedure based on multivariate polynomial approximation described in Chapter 5, Section 5.7.2, the nonlinear functions f_1 and f_2 can be reexpressed in DC form as follows

$$f_1 = g_1 - h_1, \quad f_2 = g_2 - h_2,$$

where g_1, h_1, g_2, h_2 are convex. We note that for the procedure to work, f_1 and f_2 do not need to be mathematical functions but could be defined from data. In the present case, these functions are partly defined from data through the lift and drag curves in Figure 6.2.

6.3.1 Successive convex programming tube MPC for DC systems

The nonlinear dynamics in (6.15)-(6.16) can now be expressed in a DC form as follows

$$m\dot{V}_x = g_1(V_x, V_z, i_w, T) - h_1(V_x, V_z, i_w, T) + W_x, \tag{6.19}$$

$$m\dot{V}_z = g_2(V_x, V_z, i_w, T) - h_2(V_x, V_z, i_w, T) + W_z, \tag{6.20}$$

and the DC-TMPC algorithm presented in Chapter 5 can be applied to the system.

In what follows we will exploit the convexity properties of the functions g_1, h_1, g_2, h_2 in (6.19)-(6.20) to approximate the dynamics by a set of convex inequalities with tight bounds on the state trajectories. To do so, we linearise the dynamics successively around feasible guessed trajectories and treat the linearisation error as a bounded disturbance, see Chapter 5. We use the fact that the linearisation error of a convex (resp. concave) function is also convex (resp. concave) and can thus be bounded tightly. This allows us to construct a robust optimisation using the tube-based MPC framework [69], and to obtain solutions that are robust to the model error introduced by the linearisation (i.e. model uncertainty) and to wind gusts (i.e. exogenous additive disturbances).

The DC-TMPC framework is based on the following ingredients:

Parameterisation of the control input

We start by assuming the following two-degree of freedom parameterisation of the control inputs as follows [69]

$$i_w = \mu + K_{i_w}(V_x - V_x^\circ) + K'_{i_w}(V_z - V_z^\circ),$$
$$T = \tau + K_T(V_x - V_x^\circ) + K'_T(V_z - V_z^\circ),$$

where V_x°, V_z° are guess trajectories for the states, μ, τ are feedforward terms (solution of the MPC optimisation stated in Section 6.3.2) and $K_{i_w}, K'_{i_w}, K_T, K'_T$ are feedback gains to be computed e.g. by solving a LQR problem for the linearised nominal ($W_x, W_z = 0$) system (6.19)-(6.20). Note that g_1, h_1, g_2, h_2 defined in (6.19)-(6.20) are now functions of V_x, V_z, μ, τ.

Successive linearisations

We assume the existence[4] of a set of feasible guess trajectories $V_x^\circ, V_z^\circ, \mu^\circ, \tau^\circ$ for (6.19)-(6.20) and successively linearise the dynamics around the guessed trajectories[5]. The Taylor series expansion of the nonlinear dynamics is given by $\forall i \in \{1, 2\}$

$$g_i = \lfloor g_i \rfloor_{(V_x^\circ, V_z^\circ, \mu^\circ, \tau^\circ)} + \lceil g_i \rceil_{(V_x^\circ, V_z^\circ, \mu^\circ, \tau^\circ)},$$
$$h_i = \lfloor h_i \rfloor_{(V_x^\circ, V_z^\circ, \mu^\circ, \tau^\circ)} + \lceil h_i \rceil_{(V_x^\circ, V_z^\circ, \mu^\circ, \tau^\circ)},$$

where the notation $\lfloor f \rfloor_{x^\circ} = f(x^\circ) + \nabla f^\top(x^\circ)(x - x^\circ)$ stands for the Jacobian linear approximation of f around x° and $\lceil f \rceil_{x^\circ}$ the corresponding linearisation error. After each iteration of the algorithm, the guessed trajectories are updated with the solution of the MPC optimisation and a new pass is initiated by linearising the dynamics around the new estimate.

Parameterisation of the uncertainty sets

We assume that the uncertain state trajectories V_x, V_z lie within "tubes" whose cross-sections are parameterised by means of elementwise bounds $V_x \in [\underline{V}_x, \overline{V}_x]$, $V_z \in [\underline{V}_z, \overline{V}_z]$, which are optimisation variables, see Figure 6.3. Crucially these sets are low complexity polytopes.

DC properties of dynamics

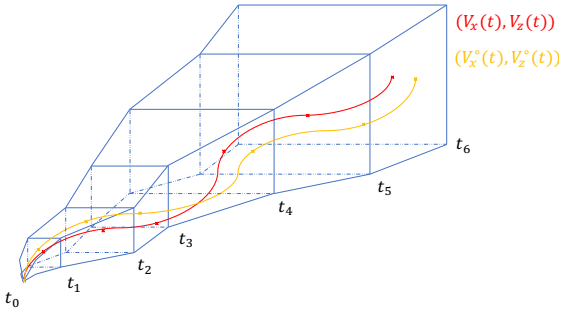
By convexity of g_1, g_2, h_1, h_2, the associated linearisation errors are necessarily convex and take their maximum at one of the vertices of the polytopic uncertainty set on which they are defined[6]. Moreover, by definition of the Jacobian linearisation, their minimum on this set is zero[7]. It follows that the bounds on the states

[4] We can obtain feasible initial trajectories by simulating the nominal aircraft dynamics with a prior-determined control law, such as PID, and checking *a posteriori* that other constraints are satisfied. An alternative method is to solve an initial feasibility problem as discussed in Chapter 5.

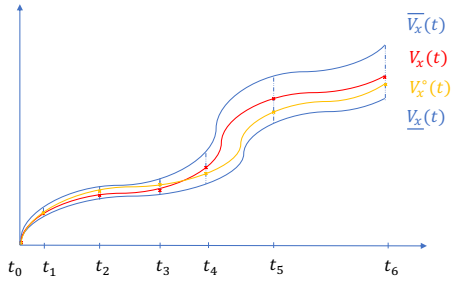
[5] According to the developments in Chapter 5, Section 5.7.1, in the presence of additive uncertainty we normally have to construct a feasibility tube and obtain the linearisation trajectories from that tube in order to guarantee recursive feasibility.

[6] Appendix A, Theorem 14 and Lemma 15.

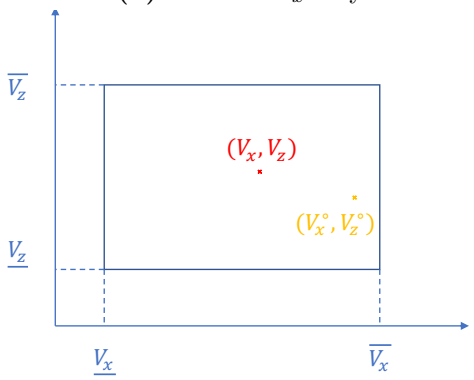
[7] Ibid., Theorem 16



(a) Tube



(b) Tube for V_x only



(c) Tube cross section

Figure 6.3: Schematic visualisation of the tube. (a) The uncertain state trajectory (red) lies within a tube (blue) centred around the nominal trajectory (yellow), sampled at various time steps. Also shown is the tube evolution for state V_x only (b) and a snapshot of the tube cross section at an arbitrary time (c).

dynamics satisfy the following convex inequalities

$$m\dot{\overline{V}_x} \geq \max_{V_x\in\{\underline{V}_x,\overline{V}_x\},V_z\in\{\underline{V}_z,\overline{V}_z\}} \Big\{g_1(V_x,V_z,\mu,\tau) - \lfloor h_1\rfloor_{(V_x^\circ,V_z^\circ,\mu^\circ,\tau^\circ)}(V_x,V_z,\mu,\tau) + \overline{W}_x\Big\}, \tag{6.21}$$

$$m\dot{\overline{V}_z} \geq \max_{V_x\in\{\underline{V}_x,\overline{V}_x\},V_z\in\{\underline{V}_z,\overline{V}_z\}} \Big\{g_2(V_x,V_z,\mu,\tau) - \lfloor h_2\rfloor_{(V_x^\circ,V_z^\circ,\mu^\circ,\tau^\circ)}(V_x,V_z,\mu,\tau) + \overline{W}_z\Big\}, \tag{6.22}$$

$$m\dot{\underline{V}_x} \leq \min_{V_x\in\{\underline{V}_x,\overline{V}_x\},V_z\in\{\underline{V}_z,\overline{V}_z\}} \Big\{-h_1(V_x,V_z,\mu,\tau) + \lfloor g_1\rfloor_{(V_x^\circ,V_z^\circ,\mu^\circ,\tau^\circ)}(V_x,V_z,\mu,\tau) + \underline{W}_x\Big\}, \tag{6.23}$$

$$m\dot{\underline{V}_z} \leq \min_{V_x\in\{\underline{V}_x,\overline{V}_x\},V_z\in\{\underline{V}_z,\overline{V}_z\}} \Big\{-h_2(V_x,V_z,\mu,\tau) + \lfloor g_2\rfloor_{(V_x^\circ,V_z^\circ,\mu^\circ,\tau^\circ)}(V_x,V_z,\mu,\tau) + \underline{W}_z\Big\}, \tag{6.24}$$

Conditions (6.21)-(6.24) must be satisfied by the tube bounding uncertain model trajectories. These constraints involve only minimisations of linear functions and maximisations of convex functions on polytopic sets. Therefore they reduce to a finite number of constraints involving the tube vertices (i.e the variables $\{\underline{V}_x,\overline{V}_x\},\{\underline{V}_z,\overline{V}_z\}$). Thus each of the constraints (6.21)-(6.24) reduces to $2^{n_x}=4$ convex inequalities ($n_x=2$).

6.3.2 Discrete time DC-TMPC

In order to obtain a finite dimensional robust MPC optimisation, we discretise the problem with a fixed sampling interval δ and evaluate all variables over a finite horizon N. The notation $\{x_0,x_1,\ldots,x_{N-1}\}$ is used for the sequence of current and future values of a variable x predicted at the n-th discrete-time step, so that x_k denotes the predicted value of $x((n+k)\delta)$.

The MPC optimisation at the n-th discrete-time step is initialised with a feasible predicted trajectory $(V_x^\circ,V_{z,k}^\circ,\mu_k^\circ,\tau_k^\circ)$ and the following optimisation problem $\mathcal{P}$ (obtained by discretising and gathering equations (6.14)-(6.18), (6.21)-(6.24)) is solved sequentially

$$\min_{\substack{\overline{V}_x,\underline{V}_x,\overline{V}_z,\underline{V}_z,\\ \tau,\mu}} \max_{\substack{V_x\in\{\underline{V}_{x,N},\overline{V}_{x,N}\},\\ V_z\in\{\underline{V}_{z,N},\overline{V}_{z,N}\}}} \left\|\begin{bmatrix}V_x-V_{x,N}^r\\ V_z-V_{z,N}^r\end{bmatrix}\right\|_{\hat{Q}}^2 + \max_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}} \left\{\sum_{k=0}^{N-1} \left\|\begin{bmatrix}V_x-V_{x,k}^r\\ V_z-V_{z,k}^r\end{bmatrix}\right\|_Q^2\right.$$

$$+\left\|\begin{bmatrix}\mu_k + K_{i_{w,k}}(V_x - V^\circ_{x,k}) + K'_{i_{w,k}}(V_z - V^\circ_{z,k}) - i^r_{w,k}\\ \tau_k + K_{T_k}(V_x - V^\circ_{x,k}) + K'_{T_k}(V_z - V^\circ_{z,k}) - T^r_k\end{bmatrix}\right\|^2_R\Bigg\},$$

s.t., $\forall k \in \{0,\dots,N-1\}$,

$$m\overline{V}_{x,k+1} \geq \max_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{mV_x + \delta g_1(V_x,V_z,\mu_k,\tau_k)$$
$$-\delta\lfloor h_1\rfloor_{(V^\circ_{x,k},V^\circ_{z,k},\mu^\circ_k,\tau^\circ_k)}(V_x,V_z,\mu_k,\tau_k) + \delta\overline{W}_{x,k}\Big\},$$

$$m\overline{V}_{z,k+1} \geq \max_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{mV_z + \delta g_2(V_x,V_z,\mu_k,\tau_k)$$
$$-\delta\lfloor h_2\rfloor_{(V^\circ_{x,k},V^\circ_{z,k},\mu^\circ_k,\tau^\circ_k)}(V_x,V_z,\mu_k,\tau_k) + \delta\overline{W}_{z,k}\Big\},$$

$$m\underline{V}_{x,k+1} \leq \min_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{mV_x - \delta h_1(V_x,V_z,\mu_k,\tau_k)$$
$$+\delta\lfloor g_1\rfloor_{(V^\circ_{x,k},V^\circ_{z,k},\mu^\circ_k,\tau^\circ_k)}(V_x,V_z,\mu_k,\tau_k) + \delta\underline{W}_{x,k}\Big\},$$

$$m\underline{V}_{z,k+1} \leq \min_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{mV_z - \delta h_2(V_x,V_z,\mu_k,\tau_k)$$
$$+\delta\lfloor g_2\rfloor_{(V^\circ_{x,k},V^\circ_{z,k},\mu^\circ_k,\tau^\circ_k)}(V_x,V_z,\mu_k,\tau_k) + \delta\underline{W}_{z,k}\Big\},$$

$$\underline{V}_{x,0} = \overline{V}_{x,0} = V_x(n\delta),\ \underline{V}_{z,0} = \overline{V}_{z,0} = V_z(n\delta),$$

$$\underline{i}_w \leq \min_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{\mu_k + K_{i_{w,k}}(V_x - V^\circ_{x,k}) + K'_{i_{w,k}}(V_z - V^\circ_{z,k})\Big\}$$

$$\bar{i}_w \geq \max_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{\mu_k + K_{i_{w,k}}(V_x - V^\circ_{x,k}) + K'_{i_{w,k}}(V_z - V^\circ_{z,k})\Big\}$$

$$\underline{V}_x \leq \underline{V}_{x,k},\ \overline{V}_{x,k} \leq \overline{V}_x,\ \underline{V}_z \leq \underline{V}_{z,k},\ \overline{V}_{z,k} \leq \overline{V}_z,$$

$$\underline{a} \leq \frac{\underline{V}_{x,k+1} - \overline{V}_{x,k}}{\delta},\ \underline{a} \leq \frac{\underline{V}_{z,k+1} - \overline{V}_{z,k}}{\delta},$$

$$\frac{\overline{V}_{x,k+1} - \underline{V}_{x,k}}{\delta} \leq \overline{a},\ \frac{\overline{V}_{z,k+1} - \underline{V}_{z,k}}{\delta} \leq \overline{a},$$

$$0 \leq \min_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{\tau_k + K_{T_k}(V_x - V^\circ_{x,k}) + K'_{T_k}(V_z - V^\circ_{z,k})\Big\},$$

$$\overline{T} \geq \max_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{\tau_k + K_{T_k}(V_x - V^\circ_{x,k}) + K'_{T_k}(V_z - V^\circ_{z,k})\Big\},$$

$$\underline{M} \leq \min_{\substack{V_x\in\{\underline{V}_{x,k},\overline{V}_{x,k}\},\\ V_z\in\{\underline{V}_{z,k},\overline{V}_{z,k}\}}}\Big\{J_w\Delta^2\{\mu_k + K_{i_{w,k}}(V_{x,k} - V^\circ_{x,k}) + K'_{i_{w,k}}(V_{z,k} - V^\circ_{z,k})\}\Big\},$$

$$\overline{M} \geq \max_{\substack{V_x \in \{\underline{V}_{x,k}, \overline{V}_{x,k}\}, \\ V_z \in \{\underline{V}_{z,k}, \overline{V}_{z,k}\}}} \Big\{ J_w \Delta^2 \{ \mu_k + K_{i_{w,k}}(V_{x,k} - V^\circ_{x,k}) + K'_{i_{w,k}}(V_{z,k} - V^\circ_{z,k}) \} \Big\},$$

$$\hat{\gamma} \geq \max_{\substack{V_x \in \{\underline{V}_{x,N}, \overline{V}_{x,N}\}, \\ V_z \in \{\underline{V}_{z,N}, \overline{V}_{z,N}\}}} \left\| \begin{bmatrix} V_x - V^r_{x,N} \\ V_z - V^r_{z,N} \end{bmatrix} \right\|^2_{\hat{Q}} \tag{6.25}$$

where $V_x(0), V_z(0)$ at time step $n = 0$ are given in Table 6.2 depending on the transition scenario considered. We denote the second order forward finite difference operator by $\Delta^2 f_k = \frac{f_{k+2} - 2f_{k+1} + f_k}{\delta^2}$. Note that the possible vertices for the tube are given by

$$\mathcal{V} = \left\{ \begin{bmatrix} \underline{V}_{x,k} \\ \underline{V}_{z,k} \end{bmatrix}, \begin{bmatrix} \overline{V}_{x,k} \\ \underline{V}_{z,k} \end{bmatrix}, \begin{bmatrix} \underline{V}_{x,k} \\ \overline{V}_{z,k} \end{bmatrix}, \begin{bmatrix} \overline{V}_{x,k} \\ \overline{V}_{z,k} \end{bmatrix} \right\},$$

which allows us to express each maximisation / minimisation above as a set of 4 inequalities at most. Moreover, since the feedback gains and the terminal penalty matrix are known *a priori*, and they are coefficients of terms that are linear in the state, this number can be further reduced to 2 for all inequalities but the first four.

Once problem $\mathcal{P}$ is solved, the guessed trajectories are updated with the solution as follows (cf. Chapter 5)

$$V_{x,0} \leftarrow V_x(n\delta), \quad V_{z,0} \leftarrow V_z(n\delta), \tag{6.26}$$

$$i^\circ_{w,k} \leftarrow \mu_k + K_{i_{w,k}}(V_{x,k} - V^\circ_{x,k}) + K'_{i_{w,k}}(V_{z,k} - V^\circ_{z,k}), \tag{6.27}$$

$$T^\circ_k \leftarrow \tau_k + K_{T_k}(V_{x,k} - V^\circ_{x,k}) + K'_{T_k}(V_{z,k} - V^\circ_{z,k}), \tag{6.28}$$

$$V_{x,k+1} \leftarrow V_{x,k} + \delta f_1(V_{x,k}, V_{z,k}, i^\circ_{w,k}, T^\circ_k)/m, \tag{6.29}$$

$$V_{z,k+1} \leftarrow V_{z,k} + \delta f_2(V_{x,k}, V_{z,k}, i^\circ_{w,k}, T^\circ_k)/m, \tag{6.30}$$

$$V^\circ_{x,k} \leftarrow V_{x,k+1}, \quad V^\circ_{z,k} \leftarrow V_{z,k+1}, \tag{6.31}$$

$$\mu^\circ_k \leftarrow \mu_k, \quad \tau^\circ_k \leftarrow \tau_k, \tag{6.32}$$

for $k = 0, \ldots N-1$ and the process of solving $\mathcal{P}$ and updating the trajectories with (6.26)-(6.32) is repeated until $||[\tau \quad \mu]^\top|| < \epsilon$. The control law at time n is then implemented by taking the first element of the control sequence

$$i_w(n\delta) = i^\circ_{w,0}, \quad T(n\delta) = T^\circ_0.$$

At time $n+1$, we set $V_{x,0} = V_x((n+1)\delta)$, $V_{z,0} = V_z((n+1)\delta)$, and update, $\forall k = 0, \ldots, N-2$ (cf. Chapter 5)

$$i^\circ_{w,k} \leftarrow i^\circ_{w,k+1},\ T^\circ_k \leftarrow T^\circ_{k+1},\ \mu^\circ_k \leftarrow \mu^\circ_{k+1},\ \tau^\circ_k \leftarrow \tau^\circ_{k+1}, \tag{6.33}$$

$$V^\circ_{x,k+1} \leftarrow V^\circ_{x,k} + \delta(f_1(V^\circ_{x,k}, V^\circ_{z,k}, i^\circ_{w,k}, T^\circ_k) + W_x)/m, \tag{6.34}$$

$$V^\circ_{z,k+1} \leftarrow V^\circ_{z,k} + \delta(f_2(V^\circ_{x,k}, V^\circ_{z,k}, i^\circ_{w,k}, T^\circ_k) + W_z)/m, \tag{6.35}$$

and finally, as per the dual mode MPC paradigm (cf. Chapter 5)

$$i^\circ_{w,N-1} \leftarrow \hat{K}_{i_w}(V^\circ_{x,N-1} - V^r_{x,N-1}) + V^r_{x,N-1} + \hat{K}'_{i_w}(V^\circ_{z,N-1} - V^r_{z,N-1}) + V^r_{z,N-1}, \tag{6.36}$$

$$T^\circ_{N-1} \leftarrow \hat{K}_T(V^\circ_{x,N-1} - V^r_{x,N-1}) + V^r_{x,N-1} + \hat{K}'_T(V^\circ_{z,N-1} - V^r_{z,N-1}) + V^r_{z,N-1}, \tag{6.37}$$

$$V^\circ_{x,N} \leftarrow V^\circ_{x,N-1} + \delta(f_1(V^\circ_{x,N-1}, V^\circ_{z,N-1}, i^\circ_{w,N-1}, T^\circ_{N-1}) + W_x)/m, \tag{6.38}$$

$$V^\circ_{z,N} \leftarrow V^\circ_{z,N-1} + \delta(f_2(V^\circ_{x,N-1}, V^\circ_{z,N-1}, i^\circ_{w,N-1}, T^\circ_{N-1}) + W_z)/m, \tag{6.39}$$

where the terminal gains $\hat{K}_{i_w}, \hat{K}'_{i_w}, \hat{K}_T, \hat{K}'_T$ can be computed following Appendix B.

6.4 Results

We consider a case study based on the transition of the Airbus A^3 Vahana (i) from powered to wing-borne flight (forward transition) and (ii) from wing-borne to powered flight (backward transition). We start with simulations conducted in the absence of wind, then consider the addition of wind disturbances. Parameters and transition boundary conditions are reported in Table 6.1 and 6.2. The terminal times for the forward and backward transitions are respectively set to $t_f = 25s$ and $t_f = 17s$ and the time step is $\delta \approx 0.22s$ in both cases, resulting in respectively $N = 110$ and $N = 75$ discretisation points. Optimisation problem $\mathcal{P}$ is solved using CVXPY [143] with solver MOSEK [132].

6.4.1 DC decomposition

The DC decompositions of f_1 and f_2 are computed according to Section 5.7.2 from Chapter 5. In each case, the approximation polynomial degree $2d$ is set to 2 and the nonlinear model f is sampled at $N_s = 1\text{e}{+4}$ evaluation points. 500 random test points are then generated to obtain the results presented in Table 6.3.

Parameter	Symbol	Value	Units
Mass	m	752.2	kg
Gravity acceleration	g	9.81	$\mathrm{m\,s^{-2}}$
Wing area	S	8.93	$\mathrm{m^2}$
Disk area	A	2.83	$\mathrm{m^2}$
Wing inertia	J_w	1100	$\mathrm{kg\,m^2}$
Density of air	ρ	1.225	$\mathrm{kg\,m^{-3}}$
Maximum thrust	$\overline{T}$	8855	N
Tiltwing angle range	$[\underline{i}_w, \overline{i}_w]$	$[-10, 100]$	deg
Acceleration range	$[\underline{a}, \overline{a}]$	$[-0.3g, 0.3g]$	$\mathrm{m\,s^{-2}}$
Forward velocity range	$[\underline{V}_x, \overline{V}_x]$	$[0, 60]$	m/s
Vertical velocity range	$[\underline{V}_z, \overline{V}_z]$	$[-30, 30]$	m/s
Torque range	$[\underline{M}, \overline{M}]$	$[-50; 50]$	N m
Number of propellers	n	4	–
Time step	δ	0.22	s
Degree of polynomial f	$2d$	2	–

Table 6.1: Model parameters derived from A^3 Vahana

Parameter	Symbol	Value	Units
Forward transition			
Forward velocity	$\{V_{x,0}; V_{x,f}\}$	$\{0; 52\}$	m/s
Vertical velocity	$\{V_{z,0}; V_{z,f}\}$	$\{0; 0\}$	m/s
Backward transition			
Forward velocity	$\{V_{x,0}; V_{x,f}\}$	$\{40; 0\}$	m/s
Vertical velocity	$\{V_{z,0}; V_{z,f}\}$	$\{0; 0\}$	m/s

Table 6.2: Boundary conditions for transitions

The least-squares mean relative error (MRE) measures how well the polynomial model $y^\top P y$ fits the nonlinear model f. The obtained errors are acceptable in the present scenario but could be further reduced if increasing the polynomial degree or using a different approximation model (e.g. radial basis functions or neural networks, see Section 5.7.2 of Chapter 5 for a presentation of various DC decomposition methods). This would typically come at the cost of increased computation times for the MPC optimisation problem. Figure 6.4 illustrates the quality of the fit for a given tiltwing angle and thrust magnitude (projection is required for visualisation purposes).

The residue of $y^\top(Q - R - P)y = 0$ illustrates the accuracy of the DC decomposition of the polynomial approximation, and is excellent in both cases.

Function	MRE (%)	Residue of $y^\top(Q-R-P)y=0$	Non PSD Hessian
f_1	5.8	5e−15	None
f_2	7.5	2e−12	None

Table 6.3: DC decomposition and least-squares fit results for 500 random test points.

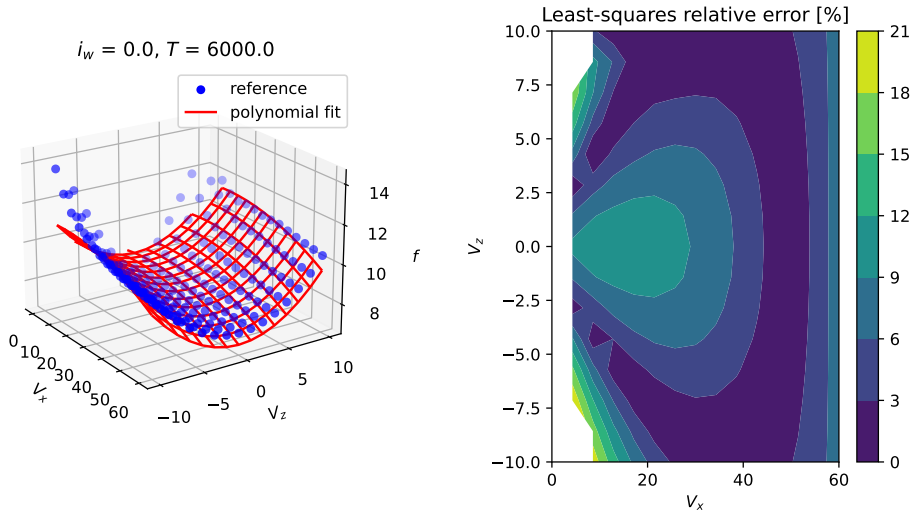


Figure 6.4: Left: least-squares fit of f_1 samples (blue dots) by the polynomial model (red curve) given i_w and T. Right: contour plot of the percent relative fitting error.

Finally, we verify that there was no convexity violation by computing the Hessians of the functions at each test point and checking for positive semidefiniteness (PSD). A typical DC decomposition is shown in Figure 6.5 (with projection).

6.4.2 Forward transition

At first, we set the state penalty matrix in the objective to $Q = \text{diag}(1, 1\text{e}{+4})$ to achieve a constant altitude forward transition. The results are shown in Figure 6.6. As the aircraft transitions from powered lift to cruise, the velocity magnitude increases, the thrust and tiltwing angle decrease, illustrating the change in lift generation from propellers to wing. The tiltwing angle drop at the beginning results in an increase in the effective angle of attack. Note that the solution (plain blue) has converged to the desired reference trajectory (dashed green) despite the initial discrepancy with the feasible guess trajectory (dashed orange).

The objective can be changed to achieve faster transitions. For example, if the state penalty matrix is set to $Q = \text{diag}(100, 1)$, the obtained results are presented

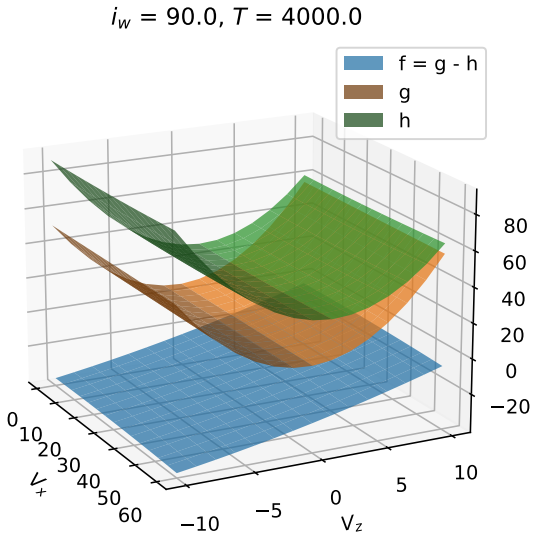


Figure 6.5: DC decomposition $f = g - h$ for given i_w and T.

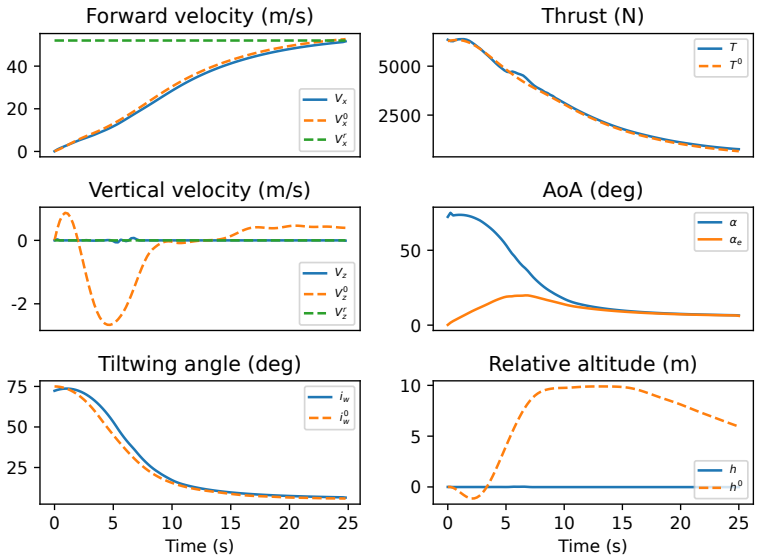


Figure 6.6: Constant altitude forward transition.

in Figure 6.7. The reference forward velocity is achieved faster than previously, but this comes at the expense of an altitude drop. A trade-off between both objectives (reaching the desired forward or vertical velocity) can be achieved by varying the penalty matrix.

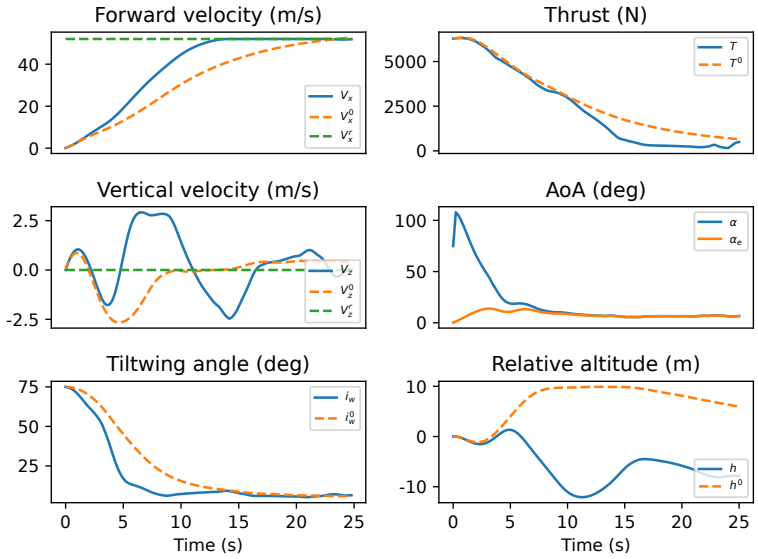


Figure 6.7: A faster forward transition.

6.4.3 Backward transition

For completeness, we consider the scenario consisting of a backward transition with an increase in altitude, see Figure 6.8. This is characterised by a decrease in velocity magnitude and increase in thrust to support the powered flight mode (hover). An increase in altitude of about 75 m is needed for this manoeuvre, and the wing is stalled.

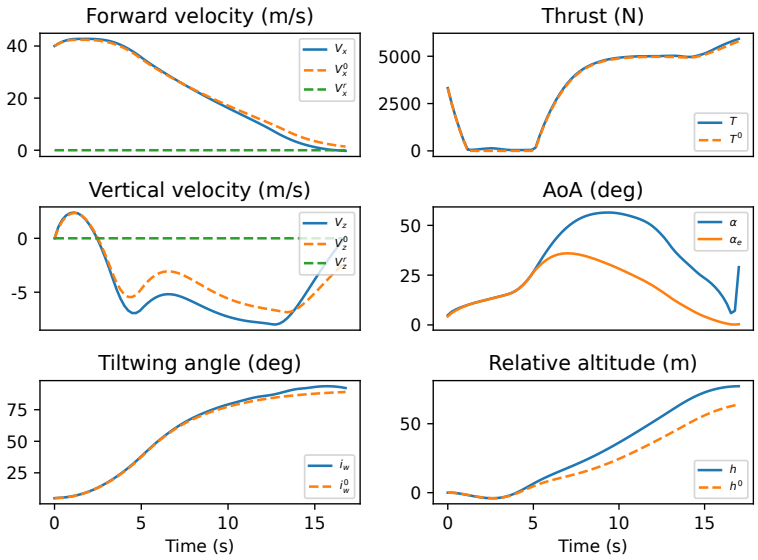


Figure 6.8: Backward transition.

6.4.4 Robustness to wind

To simulate the effect of wind gust on the aircraft, we consider EASA "Means of Compliance with the Special Condition VTOL", §2215 on flight load conditions [144] and assume that the aircraft is subject to a discrete wind gust with velocity U following a "one-minus-cosine" law

$$U(x_g) = \frac{U_{de}}{2}\left(1 - \cos\left(\frac{2\pi x_g}{25\bar{c}}\right)\right),$$

where $0 \leq x_g \leq 25\bar{c}$ is the distance penetrated into the gust, $\bar{c}$ is the mean geometric chord of the wing, and U_{de} the design gust velocity. The wind gust parameters are reported in Table 6.4 and the gust velocity profile with these values is presented in Figure 6.9.

It is assumed that the wind gust velocity acts normally to the aircraft flight path (velocity vector), i.e. along $\vec{L}$ in Figure 6.1. This is the worst case scenario for a given wind velocity magnitude as it has the effect of modifying the velocity and angle of attack seen by the wing and hence the lift and drag as follows

$$L = \tfrac{1}{2}\lambda\rho S C_L(\alpha'_e)V'^2_e + \tfrac{1}{2}(1-\lambda)\rho S C_L(\alpha')V'^2,$$
$$D = \tfrac{1}{2}\lambda\rho S C_D(\alpha'_e)V'^2_e + \tfrac{1}{2}(1-\lambda)\rho S C_D(\alpha')V'^2,$$

where

$$V'^2 = V^2 + U(x_g)^2, \quad \alpha' = \alpha + \arctan\left(\frac{U}{V}\right),$$
$$V'^2_e = V_e^2 + U(x_g)^2, \quad \alpha'_e = \alpha_e + \arctan\left(\frac{U}{V_e}\right).$$

The torque created by the imbalance in lift due to the depth difference along the wing is assumed to be negligible, which justifies our assumption that no wind gust disturbance acts on the tiltwing rotational dynamics in equation (6.4).

To evaluate the time varying wind gust bounds $[\underline{W}_i(t), \overline{W}_i(t)]$, $\forall i \in \{x, z\}$ we consider the maximum increment in drag and lift along the guess trajectory as follows

$$\Delta L_{\max}(t) = \tfrac{1}{2}\rho S C_L(\alpha^\circ)U_{de}^2 + \tfrac{1}{2}\rho S b_1 \arctan\left(U_{de}/V^\circ\right)(U_{de}^2 + V^{\circ^2}),$$

$$\Delta D_{\max}(t) = \tfrac{1}{2}\rho S C_D(\alpha^\circ) U_{de}^2 + \tfrac{1}{2}\rho S(a_2(\arctan\left(U_{de}/V^\circ\right)^2 + 2\alpha^\circ \arctan\left(U_{de}/V^\circ\right)) + a_1 \arctan\left(U_{de}/V^\circ\right))(U_{de}^2 + V^{\circ^2}),$$

where b_1, a_1, a_2 are lift and drag coefficients estimated from the curves in Figure 6.2.

Parameter	Symbol	Value	Units
Design gust velocity	U_{de}	9.14	m/s
Mean geometric chord	$\bar{c}$	1	m

Table 6.4: Wind gust parameters as defined in [144].

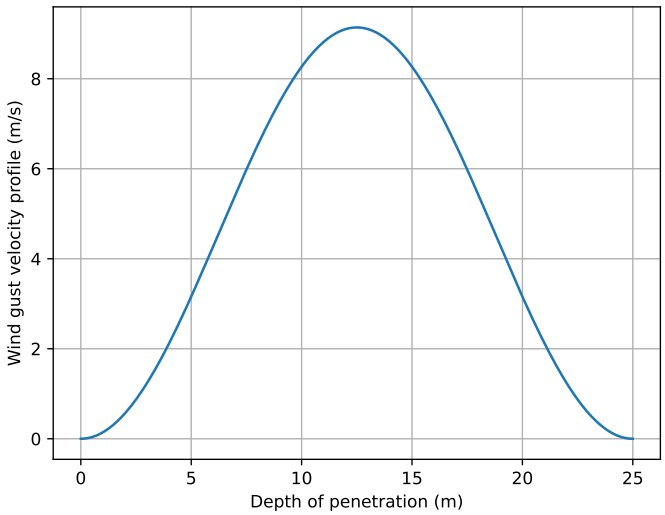


Figure 6.9: Wind gust velocity profile.

In order to evaluate the effect of wind gusts on the aircraft, we conduct multiple simulations by varying the instant at which the aircraft encounters a wind gust during the forward and backward transitions and we observe the subsequent deviations from the reference:

- **Forward transition.** The results are illustrated in Figure 6.10. For the wind gusts occuring at times $t = 5$ s and $t = 10$ s, the deviations observed are reasonable with vertical velocity not exceeding 3 m/s in magnitude. The deviation is more important when the disturbance occurs at $t = 0$ s since the vehicle is in hover mode, but the system eventually recovers and stabilises.

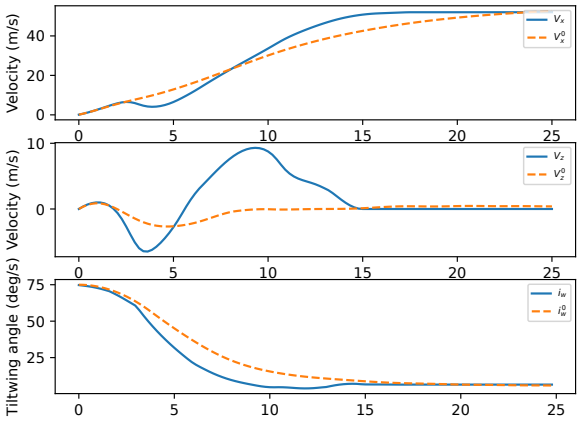
- **Backward transition.** The backward transition could not be achieved with gusts of 9.14 m/s, so the wind speed was reduced to 5.14 m/s to obtain the results in Figure 6.11. In all cases, the forward and vertical velocities are slightly perturbed and the system eventually recovers.

6.4.5 Convergence

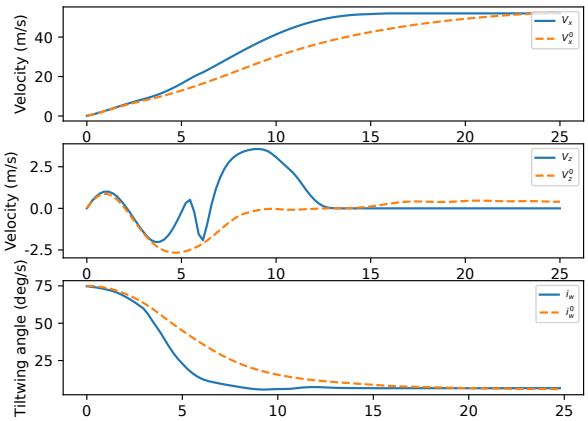
Convergence of the algorithm is illustrated in Figure 6.12, showing that the objective value decreases at each time step. Finally, Figure 6.13 shows the average computation time to solve problem $\mathcal{P}$ as a function of the number of discretisation points N. The experiment was conducted on a MacBook Pro with a 2.9 GHz dual-core Intel Core i7 processor (mid-2012). For example, for $N = 100$, the average computation time was $1.9s$. Although this would not allow to compute the solution within the specified time step in real time, it should be noted that CVXPY is not optimised for performance and that reductions in computation times of about an order of magnitude can be expected with first order solvers such as the alternating descent method of multipliers (ADMM) (cf. Chapter 3). This is in stark contrast to state-of-the-art generic NLP approaches that quote computation times of the order of minutes to solve similar VTOL transition optimisation problems (e.g. see [62]).

6.5 Conclusions

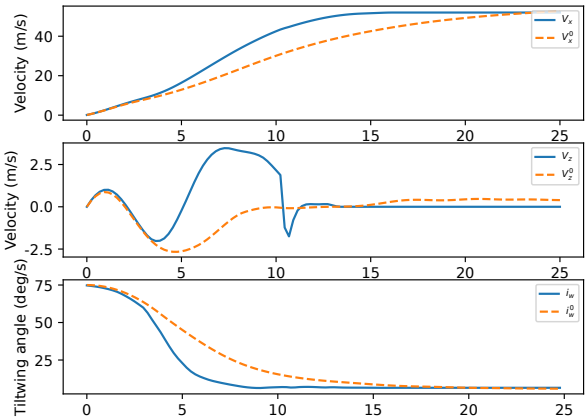
We presented a novel computationally tractable robust data-driven tube MPC scheme based on a DC decomposition of the nonlinear dynamics of a tiltwing VTOL aircraft to achieve robust transitions in the presence of wind. The DC structure of the dynamics allowed us to express the MPC optimisation at each time step as a sequence of convex programs generated by successively linearising the system around guess trajectories and bounding tightly the effect of the necessarily convex linearisation errors. We demonstrated the viability of the scheme by considering a case study inspired from the Airbus Vahana A^3 VTOL aircraft using a mixture



(a) $t = 0s$

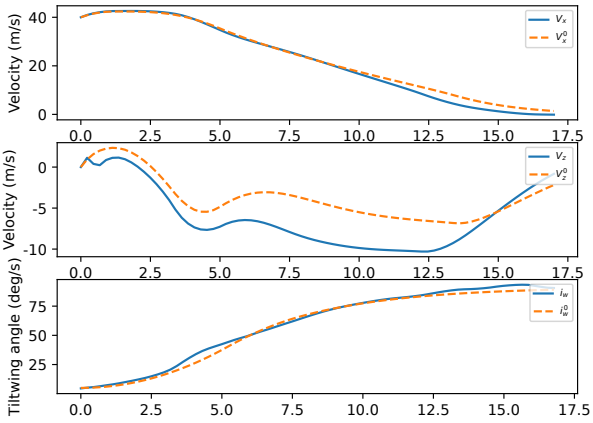


(b) $t = 5s$

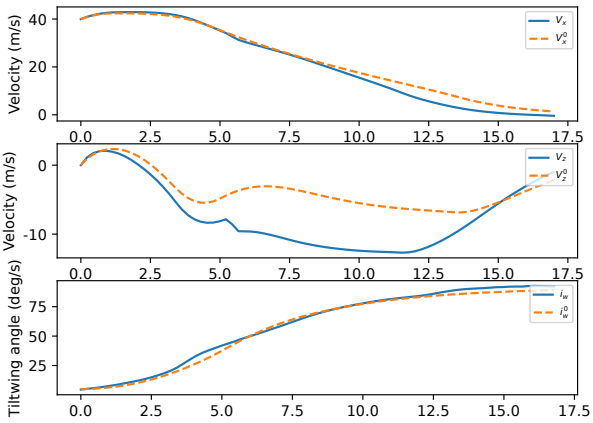


(c) $t = 10s$

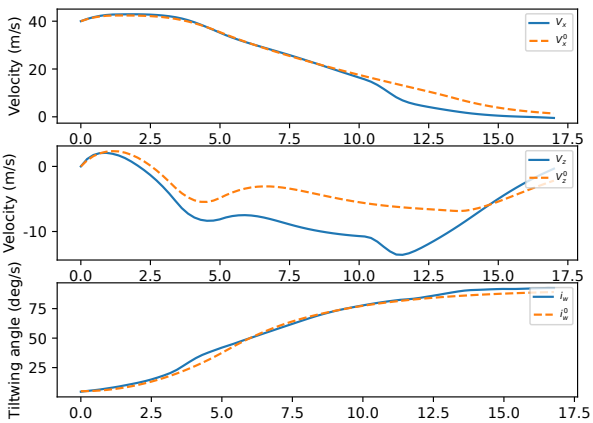
Figure 6.10: Forward transition with gust occurring at various times.



(a) $t = 0s$



(b) $t = 5s$



(c) $t = 10s$

Figure 6.11: Backward transition with gust occurring at various times.

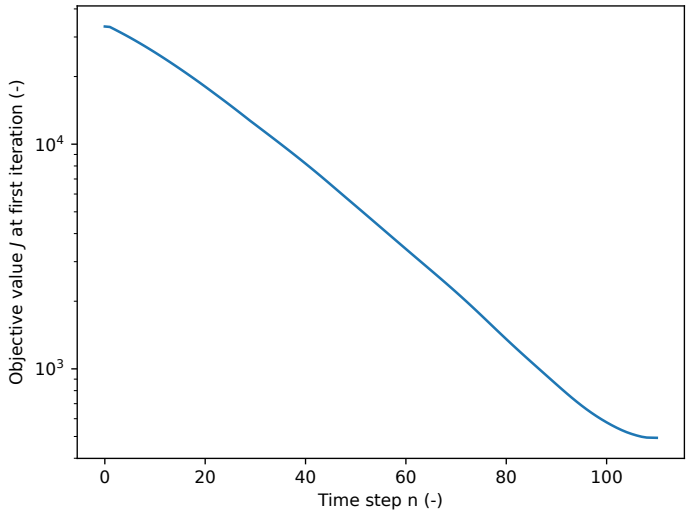


Figure 6.12: Objective at each time step.

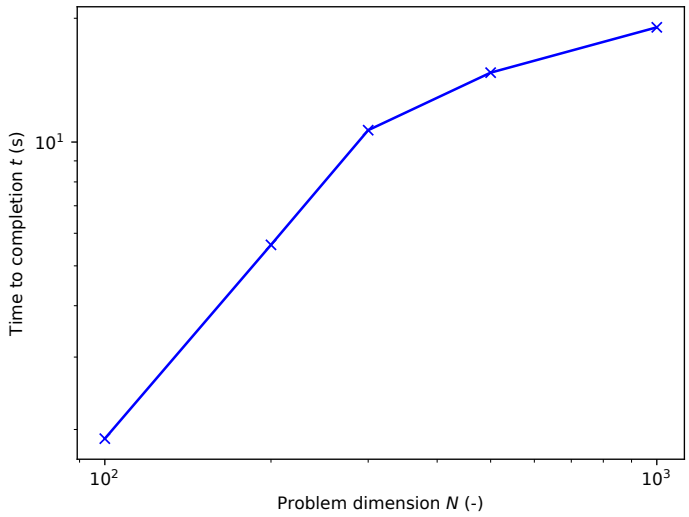


Figure 6.13: Time to completion as a function of problem dimension.

of data-based and mathematical models. Forward and backward transitions were successfully achieved, as well as transitions subject to wind gusts.

The specific contributions of this chapter are as follows: i) we propose a computationally tractable, optimal, robust control architecture for VTOL aircraft subject to additive disturbance and model uncertainty; ii) we combine DC decomposition with robust tube-based MPC and demonstrate the applicability and generalisability

of the DC-TMPC algorithm in 5; iii) we show that our technique also applies when parts of the model are defined from data.

Future work has been identified as follows: i) leveraging first order solvers (e.g. ADMM) to reduce computation times and enable real-time implementations; ii) complete study of the effect of: uncertainty set parameterisation, DC decomposition technique, problem dimension, etc. on the performance of the algorithm; iii) adaptation of the method to function approximation via a deep neural network to allow for a higher degree of generalisability (see Chapter 5, 5.7.2); iv) extension of the framework to constraints of stochastic nature (e.g. von Kármán wind turbulence model could be leveraged to achieve VTOL transitions that are less conservative); v) including in the robust optimisation an estimate of the error between the nonlinear model and the approximation function to be treated as a bounded disturbance.

7 Conclusions

Contents

7.1 Summary

This thesis investigates the development of algorithms leveraging convex optimisation, robust optimisation and model predictive control (MPC) to solve emerging problems in green aviation and urban air mobility (UAM). Chapter 3 is concerned with energy management for hybrid-electric aircraft through a convex relaxation of the optimisation in an MPC decision law. Chapter 4 introduces the idea of using difference of convex (DC) functions decompositions of the dynamics and successive linearisations to reformulate the trajectory optimisation of vertical take-off and landing (VTOL) aircraft as a sequence of computationally tractable and robust convex programs. This theory is further developed in Chapter 5, which

introduces a robust tube MPC (TMPC) algorithm for DC systems and presents computational methods for determining DC representations of smooth systems from data. Chapter 6 leverages those concepts to achieve robust stabilisation during transitions of VTOL aircraft subject to wind disturbances.

7.2 Contributions

The thesis makes contributions both in terms of aerospace applications and development of novel general control and optimisation algorithms for nonlinear systems. We detail these contributions below, organised by chapter:

- Contributions of Chapter 3:
  - Development of a convex formulation of the energy management problem for hybrid-electric aircraft in series and parallel configurations.
  - Design of a computationally tractable solver using the alternating direction method of multipliers (ADMM).
  - Use of the globally optimal solution to arbitrate in real time the power demand of the aircraft between the gas turbine and electric motor. Achievement of significant fuel savings by comparison to heuristic strategies in a short-haul flight scenario.
  - Comparison of the proposed algorithm with a general purpose convex programming solver (CVX) and investigation of the performance in terms of convergence, robustness and accuracy. Reduction of computation times by an order of magnitude.
  - Demonstration of the possibility to achieve windmilling, i.e. energy recovery during descent flight when the fan and electric motor can be operated in reverse to recharge the batteries.
- Contributions of Chapter 4:

  - Development of a new method for robust optimisation of nonlinear systems based on a difference of convex functions (DC) decomposition of the dynamics. The scheme relies on successively linearising the dynamics around predicted trajectories and treating the linearisation error as a bounded disturbance. Because the dynamics are made DC, these errors can be bounded tightly by a set of convex inequalities that form the basis of a computationally tractable robust optimisation algorithm.
  - Application of the above method to solve an open problem in trajectory optimisation of VTOL aircraft, allowing transitions at near constant altitude and high angle of attack while guaranteeing safety and computational tractability of the scheme.
  - Reduction of the problem dimension and convex relaxation of the optimisation using a change of variable and a parameterisation of the dynamics along a path.
  - Reduction of computation time by orders of magnitude compared to state-of-the-art nonlinear programming solvers for the same problem.

- Contributions of Chapter 5:

  - Development of a new robust tube-based MPC scheme for nonlinear systems that is computationally tractable, recursively feasible, converges asymptotically to a fixed point of the iteration, and guarantees closed-loop stability. The so-called DC-TMPC algorithm leverages a robust optimisation based on a DC decomposition of the dynamics and successive linearisations as part of a multi-stage robust optimal controller. Robustness to the linearisation error of the dynamics is guaranteed.
  - Crucially, the algorithm can be terminated after any number of iterations without affecting stability or constraint satisfaction, which is particularly desirable in an MPC framework.

  - Introduction of DC decomposition techniques based on polynomial basis functions, radial basis functions and neural networks, extending the scope of applicability of the theory to systems with sufficient smoothness properties (namely systems that are twice continuously differentiable).
  - Extension of the method to guarantee robustness of systems subject to additive disturbances.

- Contributions of Chapter 6:
  - Application of the DC-TMPC algorithm to solve the open problem of robust control of tiltwing VTOL aircraft subject to wind disturbances during the transition phase.
  - Demonstration of the applicability of the algorithm to solve realistic control engineering problems.
  - Demonstration of the data-driven capabilities of the algorithm, i.e. possibility to use the algorithm when parts of the model are defined from data.

7.3 Future work

We now describe potential future research directions and extensions of the theory and applications presented in this thesis. We gather these new ideas into three research projects that could be achieved in 1 to 3 years depending on the objectives set.

7.3.1 Robust optimisation in green aviation

In Chapter 3, an approach based on convex optimisation and MPC was introduced to solve the energy management problem for a hybrid-electric aircraft equipped with gas turbine and electric motor. A first extension of this work will consider other energy sources for the aircraft (e.g. hydrogen fuel cells), refining existing models of the battery or fuel cell to include thermal aspects and aging, considering the optimal deployment of supercapacitors for load smoothing, and investigating

other types of aircraft architectures, e.g. VTOL aircraft in UAM scenarios. This refinement in models and complexity will allow further energy savings, e.g. by operating subsystems at the optimal temperatures for maximum efficiency. Sophisticated nonlinear physics-based and data-based models can be incorporated without sacrificing computational tractability using DC decomposition and novel robust DC-TMPC techniques as introduced in Chapters 4-6. Since a DC reformulation can systematically be obtained for sufficiently smooth dynamics, the DC-TMPC method can be applied to a range of powertrain architectures. Another extension would be to investigate robustness of the proposed approach to power demand disturbances. Although the flight path is fixed and the aircraft flight dynamics are prescribed in the MPC optimisation, the predicted power demand is likely to be inexact and this would introduce disturbance terms into the dynamics of battery state of charge and fuel mass. Finally, the principles developed here for energy management could be applied to the problem of optimal design and sizing of the aeropropulsion components. Overall the research will contribute to green aviation by developing technologies that provide reductions of in-flight CO2 emissions in a new generation of hybrid-electric commercial aircraft.

7.3.2 Stochastic data-driven MPC of VTOL aircraft in urban air mobility scenarios

A second project will consider energy-efficient trajectory generation and robust control of a flying taxi, subject to environmental constraints of a stochastic and uncertain nature. The scenario takes place in an UAM context where space and energy resources are limited, and where issues such as congestion, obstacle avoidance (from other flying vehicles) and uncertainty (bird strike, wind) have to be factored in. VTOL aircraft capable of vertical take-off in constrained airspace and transition to energy-efficient wing-borne flight are particularly appealing solutions in this context. A multi-modal model of the environment will be built involving a mixture of mathematical models (e.g. von Kármán wind turbulence model, aircraft aerodynamics) and data-based models (weather forecast, satellite images, real-time

estimation of traffic, information from air traffic control). This model will be leveraged by the robust DC-TMPC algorithm introduced in Chapter 5 to compute energy efficient trajectories in real time and redirect the aircraft as data are collected on the fly or in the event of an obstacle (real-time collision avoidance). Due to the unpredictable environment, the guidance system should benefit greatly from the robustness capabilities of the proposed algorithm and from its computational efficiency for real-time trajectory scheduling. Moreover, the presence of data in the model will require the use of machine learning techniques such as sums of radial basis functions or neural network representations of constraints derived from multi-modal data, and will incentivise the development of *ad-hoc* DC decomposition techniques. For example, as briefly outlined in Section 5.7.2 of Chapter 5, a DC decomposition of a sum of radial basis functions can be directly obtained by enforcing simple first and second order derivative conditions, and recent developments in input-convex neural networks (ICNN) allow the construction of difference of convex neural network architectures by subtracting the output of two ICNN (with either feedforward or recurrent architecture). Such function approximations will be incorporated in the DC-TMPC formalism, showing the suitability of the method for data-driven control problems. In particular, techniques based on recurrent neural networks to approximate the system dynamics are promising due to their similarity with ordinary difference equations arising from discretisation of the system with numerical integration schemes. Issues such as quantifying the impact of uncertainty in the training data on the model parameters (e.g. the learned neural network weights) will be investigated in order to design robust controllers. For example, if the training data are corrupted by sensor noise with known probability distribution, this prior knowledge could be related to the uncertainty in the neural network weights in a Bayesian deep learning framework, and leveraged in robust optimisation approaches or controllers such as DC-TMPC to guarantee robustness to parametric uncertainty. Finally, extensions will exploit the stochastic nature of environmental disturbances (e.g. wind, stochastic model of other vehicles) in a scenario-based MPC setting. This involves generating finite sets of constraints by sampling the random variables

in a chance-constrained optimisation framework. The obtained chance-constrained optimisation will be reformulated as a sequence of convex programs using DC decomposition, with the potential to provide much less conservative bounds on the disturbances, ultimately resulting in increased energy savings. The research has the potential to revolutionise the future of mobility by accelerating the deployment of a new class of flying taxis and facilitating the adoption of explainable and robust data-driven algorithms in safety-critical applications.

7.3.3 DC-TMPC algorithms: analysis of performance and extensions

Chapter 5 has introduced a new tube-based MPC algorithm that can be applied to systems representable as a difference of convex (DC) functions. The idea is based on successive linearisations of the convex dynamics around predicted trajectories and bounding the effect of linearisation errors in a robust optimisation framework where errors are treated as additive disturbances. Because the errors are convex, they can be bounded tightly by a set of convex inequalities that form the basis of a multi-stage optimal controller named DC-TMPC. The DC-TMPC controller offers guarantees of convergence, recursive feasibility and robust stability. Crucially, it allows early termination (e.g. after a single solver iteration) without affecting these properties, which is a highly desirable characteristic of an MPC scheme. The two main limitations of the approach are: i) the *a priori* limited scope of applicability to DC systems; ii) scalability issues, typically the exponential increase of constraints with the number of states.

One of the goals of Chapter 6 was to demonstrate the viability of the method by addressing these two issues, showing that the DC-TMPC algorithm can be applied in realistic scenarios.

On the one hand, any "smooth" system can be expressed in DC form using a variety of techniques including polynomial approximation, radial basis functions, neural networks, etc. On the other hand, the computational explosion with the number of states can be mitigated using parameterisation of the uncertainty sets

with low complexity polytopes, exploiting prior knowledge of problem-specific parameters to reduce the number of inequalities when the states appear linearly, and reducing the problem dimension by other means. Interestingly, we showed in Chapter 6 that we can solve sophisticated problems with a low number of states, and that computationally tractable implementations can therefore be obtained.

This third research project aims at addressing these issues further and refining our understanding of the DC-TMPC algorithm by providing a complete analysis of performance. At first, we will discuss the influence of problem dimension on computation time and investigate ways to alleviate this problem. For example, a parameterisation of the tube in terms of simplex uncertainty sets with numbers of vertices that scale linearly with the number of states will be considered. We will in particular address whether this choice of set is overly conservative by comparison to other parameterisations (e.g. elementwise bounds, general polytopic sets, ellipsoidal sets, homothetic tube, etc.) and consider the trade-off in terms of performance.

The research will then consider DC decomposition methods related to algorithm performance. We will compare various techniques and study their influence on convergence time and accuracy. In particular we will address the problem of coping with the error introduced by the approximation in a robust control framework.

Another problem of interest will be to consider how data-based models can be integrated in the DC-TMPC framework and how uncertainty in the training data affect performance and robustness guarantees.

Finally, specialised solvers will be implemented to solve the optimisation problem efficiently with a predictable computational effort. Approaches based on the alternating direction method of multipliers (ADMM) will be considered for its potential to reduce computation times by orders of magnitude by contrast to generic convex programming solvers such as CVX.

In all cases, we will illustrate our analysis with simulation results from various case studies, e.g. N coupled tanks in series or the Fermi–Pasta–Ulam lattice system to study the effect of problem dimension; VTOL aircraft in UAM scenarios for realistic applications, etc.

Overall the research will advance the fields of robust optimisation, MPC and data-driven control by providing a systematic approach for the synthesis of computationally tractable optimisation through DC decomposition, while guaranteeing robustness and stability.

Appendices

A

Convexity Analysis

Contents

A.1 Classical results

We present here a non exhaustive list of classical results on convex functions analysis. Most results are taken from [145]. For a more exhaustive treatment, see [138].

We start by the definition of a convex set.

Definition 1 (Convex set)**.** *The set $\mathcal{S}$ is convex iff for any pair of points x, y in $\mathcal{S}$, their linear combination lies in $\mathcal{S}$*

$$\mathcal{S} \text{ is convex} \quad \Leftrightarrow \quad \lambda x + (1-\lambda)y \in S, \quad \forall \lambda \in [0,1], \quad \forall x, y \in S.$$

In other words, all line segments that start and end in a convex set are entirely contained within that set. An important type of set that we deal with in the context of tube model predictive control (TMPC) is polytopic set that we define below.

Definition 2 (Polyhedra and polytopes)**.** *A polyhedron is defined as the solution set of a finite number of linear equalities and inequalities*

$$\begin{aligned}\mathcal{P} &= \{x \,|\, a_1^\top x \leq b_1, \ldots, a_m^\top x \leq b_m\},\\ &= \{x \,|\, Ax \leq b\},\end{aligned} \tag{A.1}$$

where $A = [a_1, \ldots, a_m]^\top$, $b = [b_1, \ldots, b_m]^\top$.

A polytope (or polytopic set) is a bounded polyhedron.

Remark 8 (Convexity of polyhedra and polytopes)**.** *Polyhedra and polytopes are convex sets.*

We now recall the definition of a convex function defined over a convex set.

Definition 3 (Convex function)**.** *The function* $f : \mathrm{dom}_f \to \mathbb{R}$ *is convex iff* dom_f *is a convex set and*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \quad \forall x, y \in \mathrm{dom}_f, \quad \forall \lambda \in [0,1].$$

Remark 9 (Concave function)**.** *The function* f *is concave iff the function* $-f$ *is convex.*

We give below first order and second order conditions for convexity.

Theorem 10 (First order condition for convexity)**.** *A differentiable function* $f : \mathrm{dom}_f \to \mathbb{R}$ *is convex iff* dom_f *is a convex set and*

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x), \quad \forall x, y \in \mathrm{dom}_f.$$

In other words, the function f can be bounded from below by the Jacobian linearisation of f around any point x of its domain.

Theorem 11 (Second order condition for convexity)**.** *A twice-differentiable function* $f : \mathrm{dom}_f \to \mathbb{R}$ *is convex iff* dom_f *is a convex set and*

$$\nabla^2 f(x) \succeq 0, \quad \forall x \in \mathrm{dom}_f.$$

The above condition gives a practical tool to evaluate or enforce convexity in terms of the Hessian of the function.

We now review some of the most common convex functions

Remark 10 (Example of convex functions)**.** *The functions below are convex:*

- $f(x) = x^\alpha$, $\forall x > 0$, $\forall \alpha \geq 1$ *or* $\forall \alpha \leq 0$*;*
- $f(x) = a^T x + b$, $\forall a \in \mathbb{R}^n, b \in \mathbb{R}$ *(i.e., affine functions*[1]*);*
- $f(x) = \mathrm{e}^{\alpha x}$, $\forall \alpha \in \mathbb{R}$*;*
- $f(x, y) = x^2/y$, $\forall x, y \in \mathbb{R}^2, y \geq 0$*;*
- *any norm.*

Starting from the basic convex functions above, more complex convex functions can be constructed using operations that preserve convexity as detailed below.

Remark 11 (Convexity preserving operations)**.** *The functions f below are convex:*

- *Non-negative weighted sum:* $f(x) = \sum \alpha_i f_i(x)$ *is convex* $\forall \alpha_i \geq 0$, $\forall f_i$ *convex,* $i = 1, \ldots, m$*;*
- *Pointwise maximum:* $f(x) = \max\{f_1(x), \ldots, f_m(x)\}$ *is convex* $\forall f_i$ *convex,* $i = 1, \ldots, m$*;*
- *Composition with scalar nondecreasing functions:* $f(x) = (h \circ g)(x) = h(g(x))$ *is convex* $\forall g : \mathbb{R}^n \to \mathbb{R}$ *convex and* $\forall h : \mathbb{R} \to \mathbb{R}$ *convex nondecreasing;*
- *As a corollary of the latter:* $f(x) = g(x)^{-1}$ *is convex for concave positive* g*.*

Convex functions are important in numerical optimisation as they are the ingredients of convex programs whose solution can be obtained efficiently with modern solvers while offering guarantees of global optimality.

[1]Note that f is also concave in this particular case.

Definition 4 (Convex programs)**.** *Convex programs are of the form*

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.\ t. \quad & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & h_j(x) = 0,\ j = 1, \ldots, n \end{aligned}$$

where $f, g_j : \mathbb{R}^n \to \mathbb{R}$ *are convex and* $h_i : \mathbb{R}^n \to \mathbb{R}$ *are affine functions of* x*.*

Remark 12 (Feasibility set of convex programs)**.** *The feasible set for a convex optimisation problem is convex.*

Theorem 12 (Global optimality of convex programs)**.** *Every locally optimal solution of a convex optimisation problem is globally optimal.*

Finally, an important result due to Hartman is concerned with the decomposition of smooth functions as a difference of convex functions.

Theorem 13 (Difference of convex functions [127])**.** *Any twice-continuously-differentiable function* f *can be expressed as a difference of convex functions, i.e.*

$$f \in \mathcal{C}^2 \quad \Rightarrow \quad f = g - h,$$

where g, h *are convex functions.*

A.2 Linearisation of convex functions

We now detail a series of results which are useful to justify our approach in Chapters 4 - 6 based on bounding the linearisation errors of convex functions to construct new types of robust optimisation problems and TMPC algorithms.

Theorem 14 (Convexity of linearisation error)**.** *Let* f *be a convex function defined on a convex domain. The Jacobian linearisation error of* f *around any point* $x^\circ \in \mathrm{dom}_f$ *is also a convex function on* dom_f*.*

Proof. The Taylor's series expansion of f around x° is given by

$$f(x) = f(x^\circ) + \nabla f(x^\circ)^\top (x - x^\circ) + g(x, x^\circ),$$

where g is the Jacobian linearisation error. Solving for g yields

$$g(x, x^\circ) = f(x) - f(x^\circ) - \nabla f(x^\circ)^\top (x - x^\circ).$$

Noting that $-f(x^\circ) - \nabla f(x^\circ)^\top (x - x^\circ)$ is affine (and hence convex), it follows by convexity of f that g is also convex since it is a sum of convex functions (see Remark 11). □

The convexity property of the linearisation error is of great interest to bound its effect over a convex set. We first establish an intermediate result that states that any convex function defined on a polytopic set takes its maximum at one of its vertices.

Lemma 15 (Maximum of a convex function over a polytope)**.** *The maximum of a convex function f defined over a polytopic set $\mathcal{P} = \mathrm{Co}\{v_1, ..., v_m\}$ is achieved at one of its vertices $v_1, \ldots, v_m$, i.e.*

$$\max_{x \in \mathcal{P}} f(x) = \max\{f(v_1), \ldots, f(v_m)\}.$$

Proof. Since $\mathcal{P} = \mathrm{Co}\{v_1, ..., v_m\}$, any $x \in \mathcal{P}$ can be expressed as a convex combination of the vertices as follows $x = \sum_{i=1}^m \lambda_i v_i$ for some positive λ_i that form a partition of unity. By convexity of f, we have

$$f(x) \leq \sum_{i=1}^m \lambda_i f(v_i) \leq \max_i \{f(v_i)\},$$

which is valid $\forall x \in \mathcal{P}$ and achieves the proof. □

We are now in a position to state the main result of this section that provides bounds for the linearisation error of a convex function.

Theorem 16 (Tight bounds on convex linearisation error)**.** *Let f be a convex function defined over a convex polytopic set $\mathcal{P}$. Let g be the Jacobian linearisation error of f around $x^\circ \in \mathcal{P}$. Then:*

(i) The linearisation error g is convex;

(ii) The linearisation error g can be bounded as follows

$$0 \leq g(x, x^\circ) \leq \max_{v \in \mathcal{V}(\mathcal{P})} g(v, x^\circ),$$

where $\mathcal{V}(\mathcal{P})$ is the set of vertices of $\mathcal{P}$.

Proof. (i) From Remark 8, the polytopic set $\mathcal{P}$ is a convex set and thus the claim follows from Theorem 14.

(ii) Consider the Taylor's series expansion of f around x° as follows

$$f(x) = f(x^\circ) + \nabla f(x^\circ)^\top (x - x^\circ) + g(x, x^\circ). \tag{A.2}$$

The lower bound follows from the first order convexity condition (see Theorem 10)

$$f(x) \geq f(x^\circ) + \nabla f(x^\circ)^\top (x - x^\circ),$$

which implies $g(x, x^\circ) \geq 0$, by substitution of f in equation A.2.

The upper bound is a direct consequence of Lemma 15: since g is a convex function defined over a polytopic set, it achieves its maximum at one of the vertices and thus $g(x, x^\circ) \leq \max_{v \in \mathcal{V}(\mathcal{P})} g(v, x^\circ)$. □

B

Terminal parameters computation

We summarise here a method for computing the terminal gain $\hat{K}$, terminal weighting matrix $\hat{Q}$, and terminal set bound $\hat{\gamma}$ for the MPC problem of Chapter 5 by solving a semidefinite program (SDP). Given bounds $\bar{\mathcal{X}} = \{x : |x - x^r| \leq \delta^x\} \subseteq \mathcal{X}$, $\bar{\mathcal{U}} = \{u : |u - u^r| \leq \delta^u\} \subseteq \mathcal{U}$ on the state and control input within the terminal set, we assume that the nonlinear system dynamics can be represented in $\bar{\mathcal{X}} \times \bar{\mathcal{U}}$ using a set of linear models. The model approximation is assumed to satisfy, for all k,

$$\begin{aligned} f(x,u) - f(x^r,u^r) \in \mathrm{Co}\{&A^{(i)}(x - x_k^r) + B^{(i)}(u - u_k^r), \\ &i = 1,\ldots,m\},\ \forall (x,u) \in \bar{\mathcal{X}} \times \bar{\mathcal{U}} \end{aligned} \tag{B.1}$$

(where Co denotes the convex hull). In order that $\hat{Q}$ and $\hat{K}$ satisfy the inequality (5.5) we require, for all $x \in \bar{\mathcal{X}}$,

$$\|x - x^r\|_{\hat{Q}}^2 \geq \left\|A^{(i)}(x - x^r) + B^{(i)}\hat{K}(x - x^r)\right\|_{\hat{Q}}^2 + \|x - x^r\|_Q^2 + \|\hat{K}(x - x^r)\|_R^2.$$

Since each term is quadratic in $x - x^r$, this condition is equivalent to a set of matrix inequalities, for $i = 1,\ldots,m$,

$$\hat{Q} \succeq (A^{(i)} + B^{(i)}\hat{K})^\top \hat{Q}(A^{(i)} + B^{(i)}\hat{K}) + Q + \hat{K}^\top R\hat{K},$$

which can be expressed equivalently using Schur complements as LMIs in variables $S = \hat{Q}^{-1}$ and $Y = \hat{K}\hat{Q}^{-1}$:

$$\begin{bmatrix} S & (A^{(i)}S + B^{(i)}Y)^\top & S & Y^\top \\ \star & S & 0 & 0 \\ \star & \star & Q^{-1} & 0 \\ \star & \star & \star & R^{-1} \end{bmatrix} \succeq 0,\ i = 1, \ldots, m. \tag{B.2}$$

To ensure that the model approximation (B.1) remains valid we can exploit the positive invariance of the set $\hat{\mathcal{X}} = \{x : \|x - x^r\|_{\hat{Q}} \leq \hat{\gamma}\}$ for all $\hat{\gamma} > 0$, and impose the constraints

$$\{x : \|x - x^r\|^2_{\hat{Q}} \leq \hat{\gamma}\} \subseteq \bar{\mathcal{X}} \cap \{x : Kx \in \bar{\mathcal{U}}\}$$

which are equivalent to

$$\hat{\gamma}^{-1}[\delta^x]_i^2 - [S]_{ii} \succeq 0, \qquad i = 1, \ldots, n_x \tag{B.3}$$

$$\begin{bmatrix} \hat{\gamma}^{-1}[\delta^u]_i^2 & [Y]_i \\ \star & S \end{bmatrix} \succeq 0, \qquad i = 1, \ldots, n_u \tag{B.4}$$

To balance the requirements for good terminal controller performance and a large terminal set, we can minimise $\mathrm{tr}(\hat{Q}) + \alpha\hat{\gamma}^{-1}$ subject to the constraints (B.2), (B.3), (B.4) and

$$\begin{bmatrix} S & I \\ \star & \hat{Q} \end{bmatrix} \succ 0, \tag{B.5}$$

over variables $S = S^\top$, Y and $\hat{\gamma}^{-1}$, where α is a scalar constant that controls the trade-off between the competing objectives of minimising $\mathrm{tr}(\hat{Q})$ and minimising $\hat{\gamma}^{-1}$.

References

[1] Hoesung Lee et al. “AR6 Synthesis Report: Climate Change 2023”. In: *Summary for Policymakers* (2023).

[2] V Masson-Delmotte et al. *Global warming of 1.5∘ C. Intergovernmental Panel on Climate Change*. 2018.

[3] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Optimal energy management for hybrid electric aircraft”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 6043–6049.

[4] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Predictive energy management for hybrid electric aircraft propulsion systems”. In: *IEEE Transactions on Control Systems Technology* 31.2 (2022), pp. 602–614.

[5] Martin Doff-Sotta, Mark Cannon, and James Richard Forbes. “Spacecraft energy management using convex optimisation”. In: *Energy Conversion and Management: X* 16 (2022), p. 100325.

[6] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Fast optimal trajectory generation for a tiltwing VTOL aircraft with application to urban air mobility”. In: *2022 American Control Conference (ACC)*. IEEE. 2022, pp. 4036–4041.

[7] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Robust trajectory optimisation for transitions of tiltwing VTOL aircraft”. In: *2023 Conference on Control Technology and Applications (CCTA)*. IEEE. 2023.

[8] Martin Doff-Sotta and Mark Cannon. “Difference of convex functions in robust tube nonlinear MPC”. In: *2022 Conference on Decision and Control (CDC)*. IEEE. 2022.

[9] Niels Krausch et al. “Handling nonlinearities and uncertainties of fed-batch cultivations with difference of convex functions tube MPC”. In: *arXiv preprint arXiv:2312.00847* (2023).

[10] Johannes Buerger, Mark Cannon, and Martin Doff-Sotta. “Safe Learning in Nonlinear Model Predictive Control”. In: (2024).

[11] Martin Doff-Sotta, Mark Cannon, and Marko Bacic. “Data-driven robust model predictive control of tiltwing vertical take-off and landing aircraft”. In: *arXiv preprint arXiv:2308.03557* (2023), arXiv–2308.

[12] Bethan Owen, David S Lee, and Ling Lim. *Flying into the future: aviation emissions scenarios to 2050*. 2010.

[13] G. Hyslop et al. “Joint Statement by 7 CTOs”. In: *Paris Airshow*. 2019.

[14] Jason Welstead and James L Felder. “Conceptual design of a single-aisle turboelectric commercial transport with fuselage boundary layer ingestion”. In: *54th AIAA Aerospace Sciences Meeting*. 2016.

[15] Wim Lammen and Jos Vankan. "Energy Optimization of Single Aisle Aircraft with Hybrid Electric Propulsion". In: *AIAA Scitech 2020 Forum*. 2020.

[16] Kevin R Antcliff and Francisco M Capristan. "Conceptual design of the parallel electric-gas architecture with synergistic utilization scheme (PEGASUS) concept". In: *18th AIAA/ISSMO multidisciplinary analysis and optimization conference*. 2017, p. 4001.

[17] ICAO Secretariat. "Electric, Hybrid, and Hydrogen Aircraft–State of Play". In: *Climate Change Mitigation: Technology and Operations* (2019), pp. 124–130.

[18] Peter Schmollgruber et al. "Multidisciplinary design and performance of the ONERA Hybrid Electric Distributed Propulsion concept (DRAGON)". In: *AIAA Scitech 2020 Forum*. 2020.

[19] Alex M Stoll et al. "Drag reduction through distributed electric propulsion". In: *14th AIAA aviation technology, integration, and operations conference*. 2014.

[20] Matthias Strack et al. "Conceptual Design Assessment of Advanced Hybrid Electric Turboprop Aircraft Configurations". In: *17th AIAA Aviation Technology, Integration, and Operations Conference*. 2017, p. 3068.

[21] Bulent Sarlioglu and Casey T Morris. "More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft". In: *IEEE transactions on Transportation Electrification* 1.1 (2015), pp. 54–64.

[22] Souleman Njoya Motapon, Louis-A Dessaint, and Kamal Al-Haddad. "A comparative study of energy management schemes for a fuel-cell hybrid emergency power system of more-electric aircraft". In: *IEEE transactions on industrial electronics* 61.3 (2013), pp. 1320–1334.

[23] Airbus. *ZEROe: Towards the world's first zero-emission commercial aircraft*. 2019.

[24] Benjamin J Brelje and Joaquim RRA Martins. "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches". In: *Progress in Aerospace Sciences* 104 (2019), pp. 1–19.

[25] Laura Tribioli et al. "A real time energy management strategy for plug-in hybrid electric vehicles based on optimal control theory". In: *Energy Procedia* 45 (2014), pp. 949–958.

[26] Xian-Zhong Gao et al. "Energy management strategy for solar-powered high-altitude long-endurance aircraft". In: *Energy conversion and management* 70 (2013), pp. 20–30.

[27] Cosimo Spagnolo et al. "Finite state machine control for aircraft electrical distribution system". In: *The Journal of Engineering* 2018.13 (2018), pp. 506–511.

[28] S Caux et al. "On-line fuzzy energy management for hybrid fuel cell systems". In: *International journal of hydrogen energy* 35.5 (2010), pp. 2134–2143.

[29] Julian Hoelzen et al. "Conceptual design of operation strategies for hybrid electric aircraft". In: *Energies* 11.1 (2018), p. 217.

[30] Xi Zhang et al. "Wavelet-transform-based power management of hybrid vehicles with multiple on-board energy sources including fuel cell, battery and ultracapacitor". In: *Journal of Power Sources* 185.2 (2008), pp. 1533–1543.

[31] Sebastian East and Mark Cannon. "Optimal power allocation in battery/supercapacitor electric vehicles using convex optimization". In: *IEEE Trans. Veh. Technol.* 69.11 (2020), pp. 12751–12762.

[32] Jorge Moreno, Micah E Ortúzar, and Juan W Dixon. "Energy-management system for a hybrid electric vehicle, using ultracapacitors and neural networks". In: *IEEE transactions on Industrial Electronics* 53.2 (2006), pp. 614–623.

[33] Lakmal Karunarathne, John T Economou, and Kevin Knowles. "Model based power and energy management system for PEM fuel cell/Li-Ion battery driven propulsion system". In: (2010).

[34] Pablo García et al. "Viability study of a FC-battery-SC tramway controlled by equivalent consumption minimization strategy". In: *International journal of hydrogen energy* 37.11 (2012), pp. 9368–9382.

[35] Souleman Njoya Motapon, Louis-A Dessaint, and Kamal Al-Haddad. "A Robust H_2-Consumption-Minimization-Based Energy Management Strategy for a Fuel Cell Hybrid Emergency Power System of More Electric Aircraft". In: *IEEE Transactions on Industrial Electronics* 61.11 (2014), pp. 6148–6156.

[36] Thomas Bradley et al. "Energy management for fuel cell powered hybrid-electric aircraft". In: *7th international energy conversion engineering conference*. 2009, p. 4590.

[37] Min-Joong Kim and Huei Peng. "Power management and design optimization of fuel cell/battery hybrid vehicles". In: *Journal of power sources* 165.2 (2007), pp. 819–832.

[38] C. Lin et al. "Power management strategy for a parallel hybrid electric truck". In: *IEEE Transactions on Control Systems Technology* 11.6 (2003), pp. 839–849.

[39] Pierluigi Pisu and Giorgio Rizzoni. "A comparative study of supervisory control strategies for hybrid electric vehicles". In: *IEEE transactions on control systems technology* 15.3 (2007), pp. 506–518.

[40] Wei-Song Lin and Chen-Hong Zheng. "Energy management of a fuel cell/ultracapacitor hybrid power system using an adaptive optimal-control method". In: *Journal of Power Sources* 196.6 (2011), pp. 3280–3289.

[41] M Koot et al. "Energy Management Strategies for Vehicular Electric Power Systems". In: *IEEE Trans. Veh. Technol.* 54.3 (2005), pp. 771–782.

[42] Sebastian East and Mark Cannon. "Energy management in plug-in hybrid electric vehicles: Convex optimization algorithms for model predictive control". In: *IEEE Transactions on Control Systems Technology* (2019).

[43] Johannes Buerger, Sebastian East, and Mark Cannon. "Fast dual-loop nonlinear receding horizon control for energy management in hybrid electric vehicles". In: *IEEE Transactions on Control Systems Technology* 27.3 (2018), pp. 1060–1070.

[44] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[45] Alexander Domahidi et al. "Efficient interior point methods for multistage problems arising in receding horizon control". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 668–674.

[46] Xi Zhang, Laura Ferranti, and Tamás Keviczky. "An improved primal-dual interior point solver for model predictive control". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 1126–1131.

[47] Hans Joachim Ferreau, Hans Georg Bock, and Moritz Diehl. "An online active set strategy to overcome the limitations of explicit MPC". In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 18.8 (2008), pp. 816–830.

[48] Stefan Richter, Sebastien Mariethoz, and Manfred Morari. "High-speed online MPC based on a fast gradient method applied to power converter control". In: *Proceedings of the 2010 American Control Conference*. IEEE. 2010, pp. 4737–4743.

[49] Idris Kempf, Paul Goulart, and Stephen Duncan. "Fast Gradient Method for Model Predictive Control with Input Rate and Amplitude Constraints". In: *arXiv preprint arXiv:2003.05667* (2020).

[50] S. Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine learning* 3.1 (2011), pp. 1–122.

[51] S. East and M. Cannon. "An ADMM algorithm for MPC-based energy management in hybrid electric vehicles with nonlinear losses". In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 2641–2646.

[52] Jinwoo Seok, Ilya Kolmanovsky, and Anouck Girard. "Coordinated model predictive control of aircraft gas turbine engine and power system". In: *Journal of Guidance, Control, and Dynamics* 40.10 (2017), pp. 2538–2555.

[53] William Dunham et al. "Distributed Model Predictive Control for More Electric Aircraft Subsystems Operating at Multiple Time Scales". In: *IEEE Transactions on Control Systems Technology* (2019).

[54] David Schrank et al. "2015 urban mobility scorecard". In: (2015).

[55] McKinsey. "Study on the societal acceptance of Urban Air Mobility in Europe". In: *European Union Aviation Safety Agency (EASA)* (2021).

[56] Manfred Hader et al. "Urban air mobility, USD 90 billion of potential: how to capture a share of the passenger drone market". In: *Roland Berger* (2020).

[57] Akshay R Kadhiresan and Michael J Duffy. "Conceptual design and mission analysis for eVTOL urban air mobility flight vehicle configurations". In: *AIAA aviation 2019 forum*. 2019, p. 2873.

[58] Chana Anna Saias et al. "Assessment of hydrogen fuel for rotorcraft applications". In: *International Journal of Hydrogen Energy* 47.76 (2022), pp. 32655–32668.

[59] Richard E Kuhn. "Semiempirical procedure for estimating lift and drag characteristics of propeller-wing-flap configurations for vertical-and short-take-off-and-landing airplanes". In: *NASA Memorandum* (1959).

[60] Charles R Hargraves. *An analytical study of the longitudinal dynamics of a tilt-wing VTOL*. Tech. rep. Princeton University, 1961.

[61] Barnes Warnock McCormick. *Aerodynamics of V/STOL flight*. Academic Press, London, 1967.

[62] Shamsheer S Chauhan and Joaquim RRA Martins. "Tilt-wing eVTOL takeoff trajectory optimization". In: *Journal of Aircraft* (2019), pp. 1–20.

[63] Graham Warwick. "Inside Airbus A3's Vahana Electric VTOL". In: *Aviation Week & Space Technology* (2017).

[64] Paul M Rothhaar et al. "NASA Langley distributed propulsion VTOL tiltwing aircraft testing, modeling, simulation, control, and flight test development". In: *14th AIAA aviation technology, integration, and operations conference*. 2014, p. 2999.

[65] Ross J Higgins et al. "An Aeroacoustic Investigation of a Tiltwing eVTOL Concept Aircraft". In: *AIAA AVIATION 2020 FORUM*. 2020, p. 2684.

[66] Romain Chiappinelli et al. "Modeling and control of a passively-coupled tilt-rotor vertical takeoff and landing aircraft". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 4141–4147.

[67] Priyank Pradeep and Peng Wei. "Energy optimal speed profile for arrival of tandem tilt-wing eVTOL aircraft with RTA constraint". In: *IEEE CSAA Guidance, Navigation and Control Conference*. 2018.

[68] Leo Panish and Marko Bacic. "Transition Trajectory Optimization for a Tiltwing VTOL Aircraft with Leading-Edge Fluid Injection Active Flow Control". In: *AIAA Scitech 2022 San Diego* (2022).

[69] Basil Kouvaritakis and Mark Cannon. "Model predictive control". In: *Switzerland: Springer International Publishing* (2016).

[70] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[71] Mayuresh V Kothare, Venkataramanan Balakrishnan, and Manfred Morari. "Robust constrained model predictive control using linear matrix inequalities". In: *Automatica* 32.10 (1996), pp. 1361–1379.

[72] David Mayne. "Robust and stochastic model predictive control: Are we going in the right direction?" In: *Annual Reviews in Control* 41 (2016), pp. 184–192.

[73] Ali Mesbah. "Stochastic model predictive control: An overview and perspectives for future research". In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44.

[74] H Witsenhausen. "A minimax control problem for sampled linear systems". In: *IEEE Transactions on Automatic Control* 13.1 (1968), pp. 5–21.

[75] MC Delfour and SK Mitter. "Reachability of perturbed systems and min sup problems". In: *SIAM Journal on Control* 7.4 (1969), pp. 521–533.

[76] Dimitri P Bertsekas and Ian B Rhodes. "On the minimax reachability of target sets and target tubes". In: *Automatica* 7.2 (1971), pp. 233–247.

[77] Peter J Campo and Manfred Morari. "Robust model predictive control". In: *1987 American control conference*. IEEE. 1987, pp. 1021–1026.

[78] Jesse R Gossner, Basil Kouvaritakis, and John Anthony Rossiter. "Stable generalized predictive control with constraints and bounded disturbances". In: *Automatica* 33.4 (1997), pp. 551–568.

[79] YI Lee and B Kouvaritakis. "Constrained receding horizon predictive control for systems with disturbances". In: *International Journal of Control* 72.11 (1999), pp. 1027–1032.

[80] J Schuurmans and JA Rossiter. "Robust predictive control using tight sets of predicted states". In: *IEE proceedings-Control theory and applications* 147.1 (2000), pp. 13–18.

[81] Basil Kouvaritakis, John Anthony Rossiter, and Jan Schuurmans. "Efficient robust predictive control". In: *IEEE Transactions on automatic control* 45.8 (2000), pp. 1545–1549.

[82] Young Il Lee and Basil Kouvaritakis. "Robust receding horizon predictive control for systems with uncertain dynamics and input saturation". In: *Automatica* 36.10 (2000), pp. 1497–1504.

[83] Mark Cannon et al. "Robust tubes in nonlinear model predictive control". In: *IEEE Transactions on Automatic Control* 56.8 (2011), pp. 1942–1947.

[84] Paul J Goulart, Eric C Kerrigan, and Jan M Maciejowski. "Optimization over state feedback policies for robust control with constraints". In: *Automatica* 42.4 (2006), pp. 523–533.

[85] J Glover and F Schweppe. "Control of linear dynamic systems with set constrained disturbances". In: *IEEE Transactions on Automatic Control* 16.5 (1971), pp. 411–423.

[86] Dimitri P Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Academic press, 1976.

[87] Luigi Chisci, John Anthony Rossiter, and Giovanni Zappa. "Systems with persistent disturbances: predictive control with restricted constraints". In: *Automatica* 37.7 (2001), pp. 1019–1028.

[88] Wilbur Langson et al. "Robust model predictive control using tubes". In: *Automatica* 40.1 (2004), pp. 125–133.

[89] Martin Evans, Mark Cannon, and Basil Kouvaritakis. "Robust MPC for linear systems with bounded multiplicative uncertainty". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 248–253.

[90] James Fleming, Basil Kouvaritakis, and Mark Cannon. "Robust tube MPC for linear systems with multiplicative uncertainty". In: *IEEE Transactions on Automatic Control* 60.4 (2014), pp. 1087–1092.

[91] Zhong-Ping Jiang and Yuan Wang. "Input-to-state stability for discrete-time nonlinear systems". In: *Automatica* 37.6 (2001), pp. 857–869.

[92] Paul J Goulart, Eric C Kerrigan, and Daniel Ralph. "Efficient robust optimization for robust control with constraints". In: *Mathematical Programming* 114.1 (2008), pp. 115–147.

[93] David Q Mayne, María M Seron, and SV Raković. "Robust model predictive control of constrained linear systems with bounded disturbances". In: *Automatica* 41.2 (2005), pp. 219–224.

[94] Saša V Raković et al. "Homothetic tube model predictive control". In: *Automatica* 48.8 (2012), pp. 1631–1638.

[95] Young Il Lee and Basil Kouvaritakis. "A linear programming approach to constrained robust predictive control". In: *IEEE Transactions on Automatic Control* 45.9 (2000), pp. 1765–1770.

[96] L Magni et al. "Robust model predictive control for nonlinear discrete-time systems". In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 13.3-4 (2003), pp. 229–246.

[97] D Limon et al. "Robust MPC of constrained nonlinear systems based on interval arithmetic". In: *IEE Proceedings-Control Theory and Applications* 152.3 (2005), pp. 325–332.

[98] David Q Mayne and Eric C Kerrigan. "Tube-based robust nonlinear model predictive control". In: *IFAC Proceedings Volumes* 40.12 (2007), pp. 36–41.

[99] David Q Mayne et al. "Tube-based robust nonlinear model predictive control". In: *International Journal of Robust and Nonlinear Control* 21.11 (2011), pp. 1341–1353.

[100] Hanna Michalska and David Q Mayne. "Robust receding horizon control of constrained nonlinear systems". In: *IEEE transactions on automatic control* 38.11 (1993), pp. 1623–1633.

[101] Young Il Lee, Basil Kouvaritakis, and Mark Cannon. "Constrained receding horizon predictive control for nonlinear systems". In: *Automatica* 38.12 (2002), pp. 2093–2102.

[102] Matthias Althoff, Olaf Stursberg, and Martin Buss. "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization". In: *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 4042–4048.

[103] Mark Cannon, Desmond Ng, and Basil Kouvaritakis. "Successive linearization NMPC for a class of stochastic nonlinear systems". In: *Nonlinear Model Predictive Control*. Springer, 2009, pp. 249–262.

[104] Sasa V Raković, William S Levine, and Behccet Accikmese. "Elastic tube model predictive control". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 3594–3599.

[105] Brett T Lopez, Jonathan P How, and Jean-Jacques E Slotine. "Dynamic tube MPC for nonlinear systems". In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 1655–1662.

[106] Matteo Rubagotti et al. "Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems". In: *IEEE Transactions on Automatic Control* 56.3 (2010), pp. 556–570.

[107] Gian Paolo Incremona, Antonella Ferrara, and Lalo Magni. "Hierarchical model predictive/sliding mode control of nonlinear constrained uncertain systems". In: *IFAC-PapersOnLine* 48.23 (2015), pp. 102–109.

[108] SV Rakovic et al. "Simple robust control invariant tubes for some classes of nonlinear discrete time systems". In: *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE. 2006, pp. 6397–6402.

[109] Shuyou Yu et al. "Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems". In: *Systems & Control Letters* 62.2 (2013), pp. 194–200.

[110] Yiqi Gao et al. “A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles”. In: *Vehicle System Dynamics* 52.6 (2014), pp. 802–823.

[111] Sharmila Petkar et al. “Robust tube based MPC for PVTOL trajectory tracking using systems flatness property”. In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2016, pp. 1095–1101.

[112] D Limon et al. “Input-to-state stability: a unifying framework for robust model predictive control”. In: *Nonlinear model predictive control*. Springer, 2009, pp. 1–26.

[113] Florian Bayer, Mathias Bürger, and Frank Allgöwer. “Discrete-time incremental ISS: A framework for robust NMPC”. In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 2068–2073.

[114] Sergio Lucia and Benjamin Karg. “A deep learning-based approach to robust nonlinear model predictive control”. In: *IFAC-PapersOnLine* 51.20 (2018), pp. 511–516.

[115] David D Fan, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. “Deep learning tubes for tube MPC”. In: *arXiv preprint arXiv:2002.01587* (2020).

[116] Ruigang Wang and Jie Bao. “A differential Lyapunov-based tube MPC approach for continuous-time nonlinear processes”. In: *Journal of Process Control* 83 (2019), pp. 155–163.

[117] Sumeet Singh et al. “Robust online motion planning via contraction theory and convex optimization”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5883–5890.

[118] Anirudha Majumdar and Russ Tedrake. “Funnel libraries for real-time robust feedback motion planning”. In: *The International Journal of Robotics Research* 36.8 (2017), pp. 947–982.

[119] Mario E Villanueva et al. “Robust MPC via min–max differential inequalities”. In: *Automatica* 77 (2017), pp. 311–321.

[120] Brian L Stevens, Frank L Lewis, and Eric N Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[121] Lino Guzzella, Antonio Sciarretta, et al. *Vehicle propulsion systems*. Vol. 1. Springer, 2007.

[122] I.H. Abbott, A.E. Von Doenhoff, and L. Stivers Jr. *Summary of airfoil data*. Tech. rep. Langley Memorial Aeronautical Laboratory, 1945.

[123] Michael Grant and Stephen Boyd. “Graph implementations for nonsmooth convex programs”. In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Springer-Verlag Limited, 2008, pp. 95–110.

[124] Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. “SDPT3—a MATLAB software package for semidefinite programming, version 1.3”. In: *Optimization methods and software* 11.1-4 (1999), pp. 545–581.

[125] The MathWorks, Inc. *MATLAB Optimization Toolbox: fmincon*. `www.mathworks.com/help/optim/ug/fmincon.html`. 2021.

[126] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106 (2006), pp. 25–57.

[127] Philip Hartman. "On functions representable as a difference of convex functions." In: *Pacific Journal of Mathematics* 9.3 (1959), pp. 707–713.

[128] Thomas Lipp and Stephen Boyd. "Variations and extension of the convex–concave procedure". In: *Optimization and Engineering* 17.2 (2016), pp. 263–287.

[129] Ping Lu. "Convex–concave decomposition of nonlinear equality constraints in optimal control". In: *Journal of Guidance, Control, and Dynamics* 44.1 (2021), pp. 4–14.

[130] James E Bobrow, Steven Dubowsky, and John S Gibson. "Time-optimal control of robotic manipulators along specified paths". In: *The international journal of robotics research* 4.3 (1985), pp. 3–17.

[131] Amir Ali Ahmadi and Georgina Hall. "DC decomposition of nonconvex polynomials with algebraic techniques". In: *Mathematical Programming* 169.1 (2018), pp. 69–94.

[132] MOSEK ApS. *Introducing the MOSEK Optimization Suite 9.3.6*. 2021.

[133] R. Horst and N.V. Thoai. "DC Programming: Overview". In: *Journal of Optimization Theory and Applications* 103.1 (1999), pp. 1–43.

[134] Quanser. "Coupled tanks datasheet". In: (2015).

[135] Yana Lishkova and Mark Cannon. "Robust receding horizon control for convex dynamics and bounded disturbances". In: *arXiv preprint arXiv:2302.07744* (2023).

[136] Andrew R Barron. "Approximation and estimation bounds for artificial neural networks". In: *Machine learning* 14 (1994), pp. 115–133.

[137] Brandon Amos, Lei Xu, and J Zico Kolter. "Input convex neural networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 146–155.

[138] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[139] Parameswaran Sankaranarayanan and Raghunathan Rengaswamy. "CDiNN–Convex difference neural networks". In: *Neurocomputing* 495 (2022), pp. 153–168.

[140] Francois Chollet et al. *Keras*. 2015. URL: `https://github.com/fchollet/keras`.

[141] Yize Chen, Yuanyuan Shi, and Baosen Zhang. "Optimal control via neural networks: A convex approach". In: *arXiv preprint arXiv:1805.11835* (2018).

[142] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[143] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[144] EASA. "Means of Compliance with the Special Condition VTOL". In: *European Union Aviation Safety Agency* (2021).

[145] Paul Goulart. *Convex optimization*. 2020.