

Constrained Control Allocation Approaches in Trajectory Control of a Quadrotor Under Actuator Saturation

Talha Tariq

The Department of Mechanical Engineering

McGill University, Montreal

December, 2020

A thesis submitted to McGill University in partial fulfillment of the
requirements of the degree of

Master of Engineering

©Talha Tariq, 2020

Abstract

Quadrotors are highly agile vehicles that can be used to perform aggressive maneuvers. Commanding a quadrotor to perform a maneuver that is beyond the physical capabilities of its actuators leads to actuator saturation. A prolonged state of saturated actuators can cause the vehicle to behave unpredictably. This work investigates the use of constrained control allocation methods in a cascaded control structure to mitigate the adverse effects of actuator saturation. More specifically, a constrained weighted least squares approach is used in the position controller and mixer to prioritize certain control efforts while considering constraints on the actuators and, optionally, vehicle attitude. Additionally, a yaw-decoupled attitude controller is adopted to complement the control allocation method employed in the mixer. The proposed strategy offers a more comprehensive approach to addressing actuator saturation and was found to enhance the stability and tracking performance of the vehicle when compared to conventional approaches in simulation.

Furthermore, waypoints-based motion planning is also investigated for generating trajectories that avoid actuator saturation, as opposed to ‘handling’ saturation as is done by the controllers and mixer. The trajectories are designed for a ‘lawnmower maneuver’ and are initially generated using a minimum snap optimization algorithm. However, this method does not consider actuator or state constraints, and thus avoiding constraint violations is not guaranteed. To address this, differential flatness properties are used to evaluate these trajectories to determine whether actuator and state constraints are violated. Time scaling is then used to adjust the trajectory to meet the constraints. The appropriate scale needed for time scaling can be obtained analytically or iteratively. Analytical

solutions exist for constraints on the vehicle's velocity, acceleration, total thrust, and attitude. These solutions serve as excellent starting points for states and inputs that require an iterative approach.

Abrégé

Les quadrotors sont des véhicules très agiles qui peuvent être utilisés pour effectuer des manœuvres agressives. Ordonner à un quadrotor d'exécuter une trajectoire qui dépasse les capacités physiques de ses actionneurs conduit à une saturation de l'actionneur. Un état prolongé d'actionneurs saturés peut entraîner un comportement imprévisible du véhicule. Ce travail étudie l'utilisation de méthodes d'attribution de contrôle contraint dans une structure de contrôle en cascade pour atténuer les effets néfastes de la saturation des actionneurs. Plus spécifiquement, une approche des moindres carrés pondérés contraints est utilisée dans le contrôleur de position et le mélangeur pour hiérarchiser certains efforts de commande tout en tenant compte des contraintes sur les actionneurs et, éventuellement, sur l'attitude du véhicule. De plus, un contrôleur d'attitude découplé en lacet est adopté pour compléter le procédé d'attribution de commande utilisé dans le mélangeur. La stratégie proposée offre une approche plus complète pour traiter la saturation des actionneurs et s'est avérée améliorer la stabilité et les performances de suivi du véhicule par rapport aux approches conventionnelles en simulation.

En outre, la planification de mouvement basée sur les points de cheminement est également étudiée pour générer des trajectoires qui évitent la saturation des actionneurs, par opposition à la gestion de la saturation comme le font les contrôleurs et le mélangeur. Les trajectoires sont conçues pour une manœuvre de tondeuse à gazon et sont initialement générées à l'aide d'un algorithme d'optimisation de l'accrochage minimum. Cependant, cette méthode ne prend pas en compte les contraintes d'actionneur ou d'état, et ainsi éviter les violations de contraintes n'est pas garanti. Pour résoudre ce problème,

les propriétés de planéité différentielle sont utilisées pour évaluer ces trajectoires afin de déterminer si l'actionneur et les contraintes d'état sont violés. La mise à l'échelle temporelle est ensuite utilisée pour ajuster la trajectoire pour répondre aux contraintes. L'échelle appropriée nécessaire pour la mise à l'échelle du temps peut être obtenue de manière analytique ou itérative. Des solutions analytiques existent pour les contraintes sur la vitesse, l'accélération, la poussée totale et l'attitude du véhicule. Ces solutions constituent d'excellents points de départ pour les états et les intrants qui nécessitent une approche itérative.

Acknowledgements

I would like to thank my supervisor, Meyer Nahon, for guiding me throughout this journey and finding me opportunities where I couldn't have on my own. I feel that our meetings together have made me a far more critical thinker in the work that I do and beyond.

Thank you to everyone at the Aerospace Mechatronics Lab for being my unofficial therapists throughout all of this. Even though I didn't come to the lab very often, our moments together will ironically be the most memorable part of this journey for me.

Thank you to ARA Robotics for giving me the opportunity to collaborate with you and giving me a sense of pride in the work that I did.

I would especially like to thank my family. Your constant love reassured me that no matter what happened, you would always be there by my side.

Finally, thank you to COVID-19. Even though you shut down my experiments, you helped me realize that, in the end, the time you spend with friends and family matters most.

"I wish it need not have happened in my time," said Frodo. "So do I," said Gandalf, "and so do all who live to see such times. But that is not for them to decide. All we have to decide is what to do with the time that is given to us."

Table of Contents

Abstract	i
Abrégé	iii
Acknowledgements	v
List of Figures	x
List of Tables	xiv
1 Introduction	1
1.1 Literature Review	3
1.1.1 Mixer	3
1.1.2 Attitude Controller	5
1.1.3 Position Controller	6
1.1.4 Motion Planner	8
1.2 Thesis Organization	10
2 Modeling and Control	11

2.1	Modeling	11
2.1.1	Quadrotor Kinematics and Dynamics Model	11
2.1.2	Thruster Dynamics Model	16
2.1.3	Maneuver Description and Wind Disturbance Model	16
2.1.4	Simulation Parameters	20
2.2	Conventional Control System	21
2.2.1	Position Controller	21
2.2.2	Attitude Controller	24
2.2.3	Mixer	25
2.2.4	Simulations with Conventional Control System	26
3	Control System Improvements	30
3.1	Optimal Mixer	30
3.2	Decoupled Attitude Controller	35
3.2.1	Minimum Yaw Control	40
3.3	Optimal Position Controller	42
4	Motion Planning	52
4.1	Maneuver Description	52
4.2	Minimum Snap Trajectory Generation	54
4.3	Differential Flatness	57

4.3.1	Position, Velocity, and Acceleration	58
4.3.2	Thrust and Attitude	58
4.3.3	Angular Velocity	59
4.3.4	Angular Acceleration	60
4.3.5	Moments and Motor Thrusts	61
4.4	Time Scaling	61
4.5	Trajectory Generation and Scaling Example	62
4.6	Time Optimal Trajectory Generation	66
4.7	Second Trajectory Generation and Scaling Example	70
4.7.1	Time Optimal Trajectory	72
4.8	Analytical Time Scaling	75
4.8.1	Velocity	78
4.8.2	Acceleration	79
4.8.3	Total Thrust	79
4.8.4	Attitude	80
4.9	Analytical Scaling Example	81
5	Conclusions and Future Work	84
5.1	Summary of Research	84
5.2	Suggestions for Future Work	86

Bibliography	88
A Weight Selection	93
B Rate Constraints	97

List of Figures

1.1	Spiri quadrotor.	2
1.2	Cascaded control structure.	2
2.1	Quadrotor model with inertial (\mathcal{F}_i) and body (\mathcal{F}_b) frames.	13
2.2	Sine-wave maneuver in the inertial frame (top) and side force magnitude profile (bottom).	18
2.3	Motor speeds of all four motors vs. Time, using the conventional approach without actuator constraints, $F_{sf}^{max} = mg/2$ corresponding to wind speed = 7.52 m/s	27
2.4	Simulation plots using: the conventional approach without actuator constraints, wind speed = 7.52 m/s	28
2.5	Motor speeds of all four motors vs. Time, using the conventional approach with actuator constraints imposed, wind speed = 7.52 m/s	29
2.6	Simulation plots using: the conventional approach with actuator constraints imposed, wind speed = 7.52 m/s	29

3.1	Simulation plots using: conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{1, 1000, 1000, 100\}$), side force magnitude = $mg/2$, wind speed = 7.52 m/s	32
3.2	Simulation plots using conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{1, 1000, 1000, 100\}$), side force magnitude = mg , wind speed = 10.6 m/s	33
3.3	Simulation plots using conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	34
3.4	Simulation plots using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	37
3.5	The reduced attitude components of the desired and actual attitudes in Fig. 3.4b.	38
3.6	Simulation plots using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{20, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	39
3.7	Simulation plots with (bottom) and without (top) minimum yaw constraint using: conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	43
3.8	Simulation plots with (bottom) and without (top) optimal position controller ($\mathbf{W}_p = \text{diag}\{1, 1, 1\}$), decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	47

3.9	Simulation plots for optimal position controller (with different \mathbf{W}_p 's), decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	50
3.10	Simulation plots for optimal position controller (with different \mathbf{W}_p 's), decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s	51
4.1	Top view of Lawnmower maneuver.	54
4.2	Position trajectory given $T_1 = T_3 = 10s$ and $T_2 = 2s$	63
4.3	Plots obtained from differential flatness equations for the computed trajectory with 4 waypoints.	64
4.4	Plots obtained from differential flatness equations for the scaled trajectory with 4 waypoints, $s = 1.3$	65
4.5	Second segment of the maneuver with 4 waypoints, at different turn ratios.	68
4.6	Plots obtained from differential flatness equations for the scaled trajectory with 4 waypoints, $\mu = 0.7$	69
4.7	Position trajectory given $T_1 = T_3 = T_5 = T_7 = T_9 = 10s$ and $T_2 = T_4 = T_6 = T_8 = 2s$	70
4.8	Position segments corresponding to turns in the maneuver with 10 waypoints.	71
4.9	Plots obtained from differential flatness equations for the computed trajectory with 10 waypoints.	73
4.10	Plots obtained from differential flatness equations for the scaled trajectory with 10 waypoints, $s = 1.3$	74

4.11	Position segments corresponding to turns in the maneuver with 10 waypoints, $\mu = 0.7$.	76
4.12	Plots obtained from differential flatness equations for the scaled trajectory with 10 waypoints, $\mu = 0.7$.	77
4.13	Attitude vs. Time plot for the scaled trajectory with constraints on the pitch angle.	82
4.14	Plots obtained for the scaled trajectory with constraints shown.	83
A.1	Top view of the quadrotor model shown in Fig. 2.1.	94
A.2	Motor thrusts of all four motors vs. Time, using the conventional approach used in Fig. 2.3 without actuator constraints. Wind speed = 9.22 m/s .	95
B.1	Motor rotational speed rates using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .	98

List of Tables

2.1	Side force timestamps	20
2.2	Quadrotor, motor, and simulation parameters	21
2.3	Controller gains and disturbance characteristics	26
4.1	Turn ratios and total time for maneuver with 4 waypoints	67
4.2	Turn ratios and total time for maneuver with 10 waypoints	72

Chapter 1

Introduction

Quadrotors are a class of unmanned aerial vehicles (UAVs) that have seen widespread adoption in civilian and military applications due to advances in technology that have expanded their capabilities and substantially lowered their costs. Their appeal comes from the high agility that is afforded to them by their vertical take-off and landing (VTOL) capabilities, allowing them to perform services such as aerial inspection [1], surveillance [2], and mapping [3] safely and at high speeds. Unlike their fixed-wing counterparts, quadrotors (shown in Fig. 1.1) use four propeller-attached brushless DC motors as their actuators that supply thrust along one body-fixed axis. These actuators have physical constraints, such as minimum and maximum operating speeds (i.e., minimum and maximum achievable thrust). The maneuverability of quadrotors is often exploited to execute highly aggressive maneuvers which demand large thrust forces, potentially leading to actuator saturation. Actuator saturation is the state in which the quadrotor's control system demands performance from its actuators which exceed their physical constraints. This phenomenon becomes more probable in the presence of severe external disturbances such as wind gusts since the vehicle would require more actuator effort to counter these effects [4,5].



Figure 1.1: Spiri quadrotor.

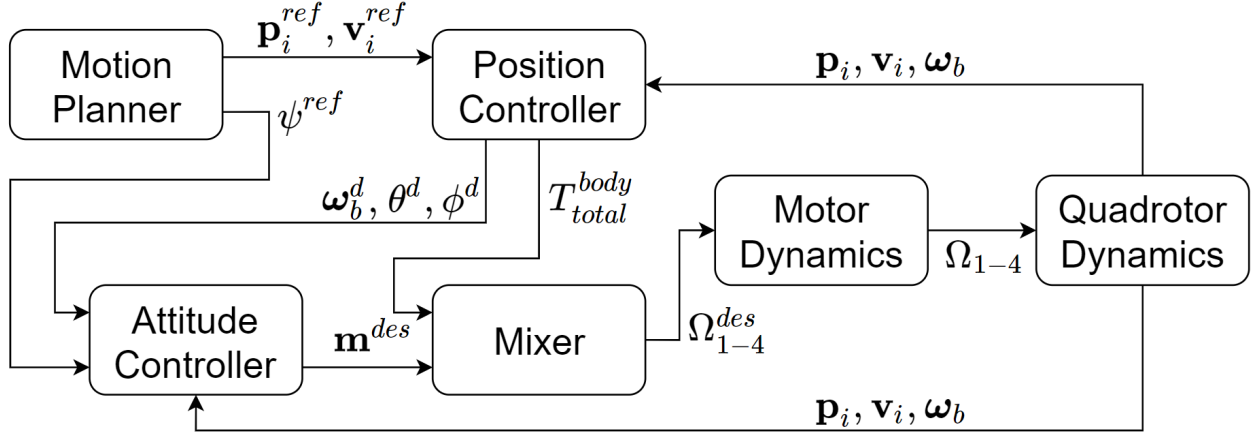


Figure 1.2: Cascaded control structure.

Conventional control strategies for trajectory-tracking that do not account for actuator saturation, or do so inadequately, can find that the vehicle behaves unpredictably under these conditions, potentially leading to a loss in performance or instability [4, 6]. Many of these control strategies are based on a cascaded control structure, shown in Fig. 1.2, examples of which can be found in [7–9]. A cascaded control structure is an appealing and effective structure for trajectory-tracking control due to the coupling that exists between a quadrotor’s translational and rotational dynamics. The coupling originates from the underactuated nature of quadrotors, a direct consequence of only being able to supply thrust along one body-fixed axis [10]. At the highest level of this cascaded structure lies the motion planner, which is responsible for generating trajectories and feeding this information to the controllers. The position controller in the outer loop is responsible for outputting desired thrusts, attitudes, and angular velocities for tracking the trajectory. In the inner loop, the attitude controller computes the necessary moments to track these desired attitudes and angular velocities. Finally, in the mixer, the control efforts from the position and attitude controllers are distributed amongst the individual actuators.

The goal of this thesis is to propose and evaluate modifications to conventional trajectory tracking approaches for position control, attitude control, and thrust mixing in order to mitigate the negative effects of actuator saturation. More specifically, these modifications will be evaluated in simulation where the stability and trajectory-tracking performance of a quadrotor are compared to those obtained with conventional approaches. Additionally, this thesis will look into the implementation of a motion planner which generates trajectories that are within the actuators' constraints. This motion planner serves a fundamentally different function from the controllers and mixer; the role of the motion planner is to generate trajectories that will avoid actuator saturation from occurring, whereas the modified controllers and mixer serve to 'handle' unexpected saturation during the maneuver most likely from unexpected disturbances. Collectively, this thesis aims to address actuator saturation at all levels of the control system.

1.1 Literature Review

This section will describe the literature pertaining to conventional approaches used in each level of the cascaded control structure and its proposed modifications. The literature review is conducted starting from the lowest level of the control structure (the mixer) and ending at the highest (the motion planner).

1.1.1 Mixer

The mixer is responsible for distributing the desired control efforts amongst the individual actuators, and this process is referred to as control allocation or thrust mixing. A thorough survey on control allocation methods is presented in [11], and [12] investigates methods that are relevant to quadrotors from this survey. In general, control allocation methods can be divided into weighted and non-weighted approaches [12]; non-weighted approaches are more commonly used for quadrotors due to their effectiveness despite

their computational simplicity. In particular, traditional control allocation, described in [4, 10, 12], has been widely used in literature and applications in which the desired thrust values are computed using a simple inverse matrix operation. This method takes advantage of the fact that, with a quadrotor platform, a determinate system of equations can be formulated in which the four control efforts are distributed amongst its four actuators. In this thesis, traditional allocation is deemed as the ‘conventional’ approach to thrust mixing due to its popularity. Other less commonly used non-weighted approaches include direct allocation, which attempts to preserve the directionality of the control effort vector [12, 13].

Non-weighted control allocation methods suffer from lacking control effort prioritization in cases where minimum and maximum thrust constraints must be considered (i.e., periods of actuator saturation). For traditional allocation, desired individual motor thrusts computed via the matrix operation are simply saturated at the minimum and maximum constraints [4, 10, 12], also referred to as thrust clipping. Consequently, neither the desired overall thrust nor moments are produced exactly by doing this; thrust clipping ignores how each control effort contributes to the stability and trajectory-tracking performance of the vehicle [4, 10]. Thus, in cases where the vehicle is in a prolonged state of actuator saturation, it would experience a loss in performance or go unstable.

Weighted control allocation methods address this issue by enabling relative control effort prioritization. In particular, numerical optimization is proposed in [6, 12, 14] where the control allocation problem is proposed as a weighted least squares problem. A weighting matrix is incorporated in the cost function of this optimization problem which can be used to prioritize certain control efforts over others. The actuator constraints can simply be considered as inequality constraints to this optimization. However, with numerical optimization comes increased computational complexity relative to non-weighted methods. Fortunately, this weighted least squares problem can be formulated as a quadratic program (QP) for which numerous efficient solvers exist for real-time applications, such as those found in [6, 14, 15]. This type of mixer is termed the ‘optimal mixer’ in this thesis.

With the optimal mixer’s weighting matrix, control effort prioritization can be set such that the mixer can relinquish yaw control in exchange for improved thrust, roll, and pitch control, the result being improved stability and performance of the vehicle [4,6,10]. More heuristic mixers are proposed in [4] and [10] to achieve the same objective while being more computationally efficient, but these are limited by the specific heuristic logic.

1.1.2 Attitude Controller

The attitude controller is the control algorithm responsible for computing the necessary (desired) moments for tracking the desired attitudes and angular velocities given by the position controller. Numerous control algorithms exist for rigid body attitude control, such as Proportional-Derivative (PD) control [7], nonlinear Model Predictive Control (MPC) [16], backstepping [9], and sliding mode control [9]. Among them, PD control has found prominence in literature and in real-life applications. In literature, it can be found in the widely cited works of [7] and [17], the latter proposing an integral term to the controller forming a PID controller. In applications, the PID controller proposed in [18] has been adopted by the PX4 Autopilot Software, one of the most popular open-source autopilot softwares currently in use amongst professionals and hobbyists.

PD (or PID) controllers are designed to track the vehicle’s attitude on the special orthogonal group $SO(3)$ and are proportional to the vehicle’s attitude and angular velocity tracking errors. However, not all PD controllers are the same; they can be differentiated by how the attitude errors are represented in their formulation. A great introduction to different attitude error representations for PD control is given in [5] and [19] with in-depth stability analysis provided for all types of PD controllers. Of these various types, the skew-symmetric PD controller is prevalent in literature due to its early adoption propelled by the works of [7,17]. This controller simply extracts the skew-symmetric component of the full attitude error as its proportional error term, parametrized using rotation matrices. In this thesis, this controller is considered as the ‘conventional’ approach to attitude control.

Using the full attitude error in this controller carries with it an unexpected consequence; in the event that there is poor yaw control leading to a significant and lingering yaw error, it indirectly causes a degradation in the roll and pitch control (collectively called reduced attitude control) [10,20]. This is due to the fact that using the full attitude error leads to a coupling between the roll and pitch error dynamics (or reduced attitude error dynamics) and yaw error dynamics. In other words, poor yaw control leads to poor reduced attitude control, which in turn leads to a degradation in the stability and performance of the quadrotor [10,20]. Clearly, this poses a problem if we were to employ the optimal mixer where yaw control is relinquished since the degradation caused by the coupling may outweigh any improvements in improved thrust, roll, and pitch control.

In [10], this issue is addressed by proposing a decoupled attitude controller to complement their heuristic formulation of an optimal mixer; the authors propose splitting and replacing the full attitude error with its reduced and yaw attitude components, with gains allocated separately for each component. In their work, the formulation is given using quaternions, and its rotation matrix-based formulation is demonstrated in [5]. This control law effectively decouples the reduced attitude error dynamics from the yaw. Decoupled attitude controllers have seen a rise in interest in recent years for this very purpose, with novel controllers being published in [5,20,21].

1.1.3 Position Controller

The position controller is responsible for computing the necessary (desired) thrusts, attitudes, and angular velocities for tracking the trajectory given by the motion planner. Numerous control algorithms exist for rigid body position control, such as Proportional-Derivative (PD) control [7], nonlinear MPC [22], and backstepping control [23]. Similar to the attitude controller, PD-based position controllers are highly popular due to their effectiveness and computational simplicity. More specifically, the works of [7] and [24] have popularized the adoption of geometric tracking control for quadrotors which utilizes PD controllers. They propose a nonlinear tracking algorithm developed on the spe-

cial Euclidean group, $SE(3)$, that is shown to have desirable closed loop properties that are almost global. The algorithm consists of a PD-based position controller as well as the aforementioned skew-symmetric (conventional) attitude controller.

The position controller can be characterized as a thrust vector tracking controller; it is proportional to the position and velocity tracking errors and outputs a desired thrust (force) vector for tracking. By constraining the vehicle's thrust vector to track the desired vector, we can compute the desired overall thrusts and attitudes needed for trajectory-tracking which is then fed to the inner loop of the control structure. However, in cases where actuator thrust constraints need to be considered, the naive approach is to simply saturate the computed overall thrust value at the constraints, similar to the conventional mixer. If the vehicle operator prefers attitude constraints as well, then the computed desired attitudes would also be subject to this saturation process. In this thesis, this position controller will be considered as the 'conventional' approach.

The authors of [25] provide an alternative to this saturation process; In [25], constrained control allocation is extended to the position controller by using least squares optimization for computing the desired overall thrusts and attitudes. This optimization problem can be formulated as another QP problem, similar to the optimal mixer, in which the actuator and attitude constraints can be added as optimization constraints. However, they differ in that a weighting matrix is not used to prioritize particular control efforts.

An enhancement of the work presented in [25] is proposed in this thesis whereby a weighting matrix is used to enable relative prioritization of control efforts in the position controller. The control efforts in this case are the components of the desired force vector computed by the PD position controller. This new position controller will be termed as the 'optimal position controller' in this thesis and will be used in combination with an optimal mixer and a yaw-decoupled attitude controller to address actuator constraints at all levels of the control system.

1.1.4 Motion Planner

The motion planner is responsible for generating trajectories and feeding this information to the controllers for tracking. Trajectories can be decomposed into segments in between waypoints, where the waypoints are specified by the vehicle operator. At minimum, there can be two waypoints for a maneuver (the start and end waypoints) for which a trajectory must be generated via the motion planner. How these waypoints are specified and under what conditions the trajectories must be generated have resulted in a diverse pool of motion planners to choose from.

In the field of motion planning for quadrotors, the works of [24] and [26] are highly regarded; the authors propose a waypoint-based trajectory generation solution that minimizes the snap (fourth derivative of position) of the vehicle given a set of waypoints with defined spatial and temporal properties. Their appeal comes from the benefit of being able to minimize the control effort needed to execute the trajectory through the minimization of the vehicle snap. Furthermore, in [24], this method can be formulated as a QP problem, similar to the optimal mixer, enabling real-time applications due to the computational efficiency of modern QP solvers. However, although these methods help generate trajectories requiring minimal control effort, their current formulation does not allow for accounting actuator constraints and thus do not guarantee avoidance of actuator saturation.

Ideally, we would like a motion planner that can generate trajectories which avoid actuator saturation and also considers constraints on the vehicle state (collectively referred to as state constraints), such as minimum/maximum attitude, velocity, acceleration etc. As previously mentioned, [24] and [26] use a QP to generate trajectories which consists of a cost function to be minimized subject to some optimization constraints. The cost function simply formulates the minimization of the vehicle snap, whereas the optimization constraints specify the waypoint locations and the boundary conditions at these waypoints. In order to consider actuator and state constraints in this optimization problem, it would be natural to consider adding them as additional optimization constraints.

In [27], the authors attempt to avoid actuator saturation by placing bounds on the total forces and moments that can be supplied by the actuators as additional optimization constraints. This approach does not guarantee saturation avoidance but does reduce the chances of saturation. In [28] and [29], maximum acceleration and attitude constraints are derived as additional optimization constraints specified at the waypoints only. Unfortunately, although these state constraints are satisfied at the waypoints, they are not guaranteed throughout the trajectory. The authors in [30] attempt to add acceleration and velocity constraints as additional terms to the cost function. In doing so, they are able to create ‘soft’ constraints for the acceleration and velocity, meaning that these constraints are satisfied to some degree of error. Using this new cost function prevents the use of a QP solver, and so a more complex nonlinear programming solver is needed which can be computationally very expensive and global convergence is not guaranteed.

Clearly, attempting to generate trajectories while considering all actuator and state constraints from a single optimization problem can be too complex for real-time applications. The authors in [24, 26, 28] suggest using the QP solver, as is, to generate an initial trajectory and then modifying it iteratively until all constraints are satisfied. To do this, we need to use the differential flatness properties of a quadrotor’s dynamics as well as the method of time scaling. Differential flatness provides an analytical mapping between the known trajectory and its derivatives to the vehicle states and control input required to track that trajectory [26]. A set of differential flatness equations have been derived for quadrotors in [31] that can be used to validate if any vehicle state or actuator input violates their constraints given any trajectory. If our initial trajectory is found to have violated a constraint via these equations, time scaling can be used to modify the trajectory’s time history until all constraints are satisfied. Time scaling is thoroughly described in [24] where the trajectory can be made more or less aggressive by scaling its parameters. If, for example, the initial trajectory was found to violate actuator constraints (i.e., it is too aggressive), then time scaling can be applied iteratively until the constraints are satisfied while using the differential flatness equations for validation.

1.2 Thesis Organization

As previously mentioned, the objectives of this thesis will be to evaluate modifications to each subsystem in a cascaded control structure which mitigate the impacts of actuator saturation. Furthermore, a motion planner will also be investigated for its capabilities in producing trajectories that avoid actuator saturation. This thesis is outlined as follows:

- The models used in simulation will be presented in Chapter 2, alongside all the conventional approaches to: position control, attitude control, and thrust mixing.
- Modifications for each of these subsystems will be described in Chapter 3, including the motivations behind them.
- A motion planner will be presented in Chapter 4, along with an introduction to differential flatness. Time scaling of the trajectories will be used to satisfy state and actuator constraints.
- The thesis is concluded and future work is suggested in Chapter 5.

Chapter 2

Modeling and Control

This chapter will describe the models used in simulation to evaluate the proposed modifications to the conventional control system. Additionally, the conventional approaches to position control, attitude control, and thrust mixing will be described here for clarity. Simulation results will also be discussed to highlight the pitfalls of using a conventional control system to conduct aggressive maneuvers in the presence of actuator saturation exacerbated by external wind disturbances. In other words, this chapter serves to provide the motivation behind the proposed modifications to the parts of the control system that will be responsible for ‘handling’ the saturation, as opposed to avoiding it (the latter being done by the motion planner).

2.1 Modeling

2.1.1 Quadrotor Kinematics and Dynamics Model

The quadrotor model formulated in [10] and [32] assumes a rigid body model for the quadrotor, and this model was adopted for the simulations conducted in this thesis. This model considers inertial, gravitational, and thrust effects, but aerodynamic drag is not

included. However, a simplified wind model was developed in which a time-varying disturbance force is exerted on the quadrotor to simulate wind disturbances.

A visual representation of the quadrotor model is shown in Fig. 2.1. The body-fixed frame is denoted by \mathcal{F}_b and its origin is coincident with the vehicle's center of mass. Its x -axis points in the vehicle's forward direction. The ground-fixed inertial frame is denoted by \mathcal{F}_i and its basis vectors along the x - y - z axes follow the North-East-Down convention, respectively.

The quadrotor is modeled as a rigid body using the equations in [10] and [32], as follows:

$$\ddot{\mathbf{p}}_i = \mathbf{C}(\mathbf{q})^T \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \frac{T_{total}}{m} + g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{F_{sf}}{m} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.1)$$

$$\dot{\boldsymbol{\omega}}_b = \mathbf{J}_b^{-1}[\mathbf{m}_b - \boldsymbol{\omega}_b \times \mathbf{J}_b \boldsymbol{\omega}_b] \quad (2.2)$$

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix} \quad (2.3)$$

where, in the translational dynamics, \mathbf{p}_i is the position vector of the vehicle's center of mass, resolved in the inertial frame (denoted by a subscript i); m refers to the mass of the vehicle, g is the acceleration due to gravity, and T_{total} is the total thrust applied by the actuators. The magnitude of the wind force exerted on the vehicle, termed the 'side force', is denoted by F_{sf} . In the rotational dynamics, $\boldsymbol{\omega}_b$ is the vehicle's angular velocity vector resolved in the body frame (denoted by a subscript b); \mathbf{J}_b and \mathbf{m}_b refer to the vehicle's second moment of mass and net external moment vector, respectively, resolved in the body frame. The Hamilton quaternion product used in the attitude kinematics is denoted by (\odot) . The attitude of the vehicle relative to the inertial frame is represented using the unit quaternion, \mathbf{q} , and $\mathbf{C}(\mathbf{q})$ is its rotation matrix equivalent. This matrix can be obtained

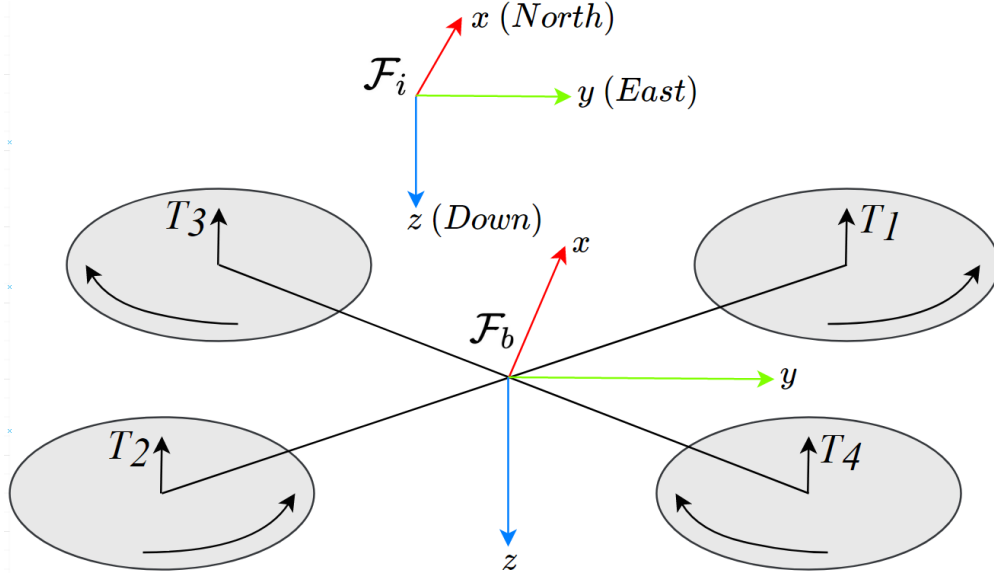


Figure 2.1: Quadrotor model with inertial (\mathcal{F}_i) and body (\mathcal{F}_b) frames.

using the following equation from [10]:

$$\mathbf{C}(\mathbf{q}) = (q_0^2 - \mathbf{q}_{1:3}^T \mathbf{q}_{1:3}) \mathbf{I} + 2(\mathbf{q}_{1:3} \mathbf{q}_{1:3}^T - q_0 \mathbf{q}_{1:3}^\times) \quad (2.4)$$

where q_0 and $\mathbf{q}_{1:3}$ denote the scalar and vector components of \mathbf{q} , respectively. The identity matrix is given by $\mathbf{I} \in \mathbb{R}^{3 \times 3}$. The cross product operator form of $\mathbf{q}_{1:3}$, $\mathbf{q}_{1:3}^\times$, is defined as:

$$\mathbf{q}_{1:3}^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (2.5)$$

The Hamilton quaternion product used in (2.3) can be expanded as follows:

$$\mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix} = \begin{bmatrix} q_1 p + q_2 q + q_3 r \\ q_0 p + q_2 r + q_3 q \\ q_0 q + q_3 p + q_1 r \\ q_0 r + q_1 q + q_2 p \end{bmatrix} \quad (2.6)$$

where p , q , and r are the x , y , and z components of $\boldsymbol{\omega}_b$, respectively.

Quaternions were selected for parametrizing the vehicle's attitude over Euler angles because Euler angles suffer from singularities at certain attitudes, referred to as 'gimbal lock' [19]. Gimbal lock is likely to occur for quadrotors since large angular rotations are expected during flight. Quaternions, on the other hand, are able to represent all possible attitudes, although the representations are not unique: each physical attitude can be represented by a pair of antipodal unit quaternions. Rotation matrices are another form of attitude parametrization that can represent the set of attitudes both globally and uniquely [19]. Nevertheless, this thesis considers controllers that use quaternions, as defined in their literature, and so quaternions were selected for kinematics modeling in simulation.

As previously stated, T_{total} is the sum of all thrusts generated by the actuators:

$$T_{total} = T_1 + T_2 + T_3 + T_4 \quad (2.7)$$

The actuators are brushless DC motors with uni-directional fixed-pitch propellers. Typically, a steady-state thruster model based on Renard's Formula is used to relate the motor's rotational speed to its generated thrust [33], as follows:

$$T_i = k_t \Omega_i^2 \quad (2.8)$$

where k_t is the propeller thrust constant and Ω_i is the rotational speed of the i^{th} motor (Ω_1 and Ω_2 are positive when the motor spins counterclockwise viewed from above; while Ω_3 and Ω_4 have the opposite convention). This model assumes that k_t remains constant and can be acquired experimentally via static testing of the motors, as is done in [34] and [35]. In reality, k_t is not a constant: the thrust generated by the thruster decreases significantly at higher vehicle speeds through air [33]. Other models have been developed to capture the effect of forward velocity on the thrust coefficient, as is done in [36] and [37]. However, in the present work, the model in (2.8) was used, with constant k_t for simplicity.

As shown in Fig. 2.1, the motors are placed at some distance away from the vehicle's center of mass. This allows the motors to generate moments about the vehicle. Using the 'X' configuration to define the body frame, these moments can be described using the equation:

$$\mathbf{m}_b^t = \begin{bmatrix} \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{k_q}{k_t} & \frac{k_q}{k_t} & \frac{-k_q}{k_t} & \frac{-k_q}{k_t} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (2.9)$$

where l is the distance between the vehicle's center of mass and each motor, and k_q is the propeller torque constant. The torque constant arises from a quadratic model analogous to that in (2.8):

$$m_{b,z_i}^t = k_q \Omega_i^2 \quad (2.10)$$

where m_{b,z_i}^t is the moment generated about the body z -axis by the i^{th} motor. Similar to k_t , k_q can be found experimentally via static testing of the motors, as is done in [34] and [35].

The thrusters also generate moments due to their angular accelerations and gyroscopic effects [34] which can be modeled using the following equation:

$$\mathbf{m}_b^m = \begin{bmatrix} qJ_r\Omega_1 + qJ_r\Omega_2 - qJ_r\Omega_3 - qJ_r\Omega_4 \\ -pJ_r\Omega_1 - pJ_r\Omega_2 + pJ_r\Omega_3 + pJ_r\Omega_4 \\ J_r\dot{\Omega}_1 + J_r\dot{\Omega}_2 - J_r\dot{\Omega}_3 - J_r\dot{\Omega}_4 \end{bmatrix} \quad (2.11)$$

where the first two elements of \mathbf{m}_b^m are due to the gyroscopic effects, and the third element is due to the angular accelerations. The thruster's moment of inertia about its thrust axis is given by J_r .

These individual moment contributions from the motors are summed to give an overall external moment applied to the quadrotor, as shown:

$$\mathbf{m}_b = \mathbf{m}_b^t + \mathbf{m}_b^m \quad (2.12)$$

2.1.2 Thruster Dynamics Model

A simple thruster dynamics model was needed to capture the transient behaviour of the thrusters. Prior works in this area have proposed a first-order model to approximate the transient behaviour [4, 35], as follows:

$$\dot{\Omega}_i = k_r(\Omega_i^{des} - \Omega_i) \quad (2.13)$$

where Ω_i^{des} is the desired RPM command of the i^{th} motor, which can be computed using (2.8) and the desired thrust command, T_i^{des} , from the mixer. The rate constant (the inverse of the time constant) is denoted by k_r . As discussed in [35], k_r can be found experimentally via static testing of the thrusters where step inputs are given to the motor and curve-fitting is applied to its response. Experimental testing shows that k_r varies with the polarity and magnitude of the term in brackets in (2.13) [4, 33]. Nonetheless, for simplicity, we assume a constant k_r in the present work.

2.1.3 Maneuver Description and Wind Disturbance Model

Maneuver Description

In order to evaluate the performance of the controllers, mixer and associated saturation handling, a maneuver had to be selected that would lead to saturation in a predictable manner. The chosen maneuver was designed to just reach saturation without any disturbances. The maneuver to be executed in simulation can be described as a sine-wave in the inertial North-East plane, as shown in Fig. 2.2. The altitude is set to be constant throughout the maneuver i.e. level flight is commanded. This type of maneuver is of particular interest because it demands enough agility from the quadrotor to challenge its capabilities and thus provide a good environment to evaluate the controller modifications proposed in this thesis. Furthermore, the maneuver is a general enough description of typical drone operations involving area coverage using onboard cameras, which will be a topic of in-

terest in Chapter 4 which involves motion planning. Level flight is commanded for this very purpose, in addition to making it easier for evaluating the impact of the proposed modifications on altitude control, as will be evident in later sections of this thesis.

The amplitude of this sine-wave is 70 *m* in the North-South (*x*) direction and ends at 100 *m* along the Eastern (*y*) direction, as shown in Fig. 2.2. The trajectory can be defined using the following equations:

$$\begin{aligned} p_{i,y}^{ref} &= \begin{cases} \frac{1}{2}a_E t^2 & \text{for } 0 \leq t < \sqrt{\frac{100}{a_E}} \\ 50 + \frac{\sqrt{100a_E}}{a_E} \left(t - \sqrt{\frac{100}{a_E}}\right) - \frac{a_E}{2} \left(t - \sqrt{\frac{100}{a_E}}\right)^2 & \text{for } t \geq \sqrt{\frac{100}{a_E}} \end{cases} \\ p_{i,x}^{ref} &= 70 \sin\left(\frac{\pi}{50}p_{i,y}^{ref}\right) \\ p_{i,z}^{ref} &= 0 \end{aligned} \quad (2.14)$$

where \mathbf{p}_i^{ref} refers to reference position vector of the trajectory (supplied by the motion planner), resolved in the inertial frame (denoted by a subscript *i*). The vehicle is initially at rest, and at the end of the maneuver \mathbf{p}_i^{ref} becomes constant to ensure that the vehicle also ends at rest. The velocity, \mathbf{v}_i^{ref} , is obtained by taking the derivative of \mathbf{p}_i^{ref} . The constant acceleration specified along the Eastern direction is denoted by a_E and is responsible for establishing the level of aggressiveness of this maneuver.

The acceleration a_E was chosen so that the vehicle would need to output maximum thrust from all of its motors near the peak and trough of the sine-wave, i.e. there will be actuator saturation. Actuator saturation is desired here for evaluating the performance of the control allocation methods. The value of a_E will depend on vehicle characteristics, like its mass, max thrust etc. so this numerical value is meaningless until we first describe the vehicle. The reference heading, ψ^{ref} , is set to be tangential to this sine-wave maneuver, as shown in Fig. 2.2, and can be defined using the equation:

$$\psi^{ref} = \frac{\pi}{2} - \arctan\left(\frac{7\pi}{5} \cos\left(\frac{\pi}{50}p_{i,y}^{ref}\right)\right) \quad (2.15)$$

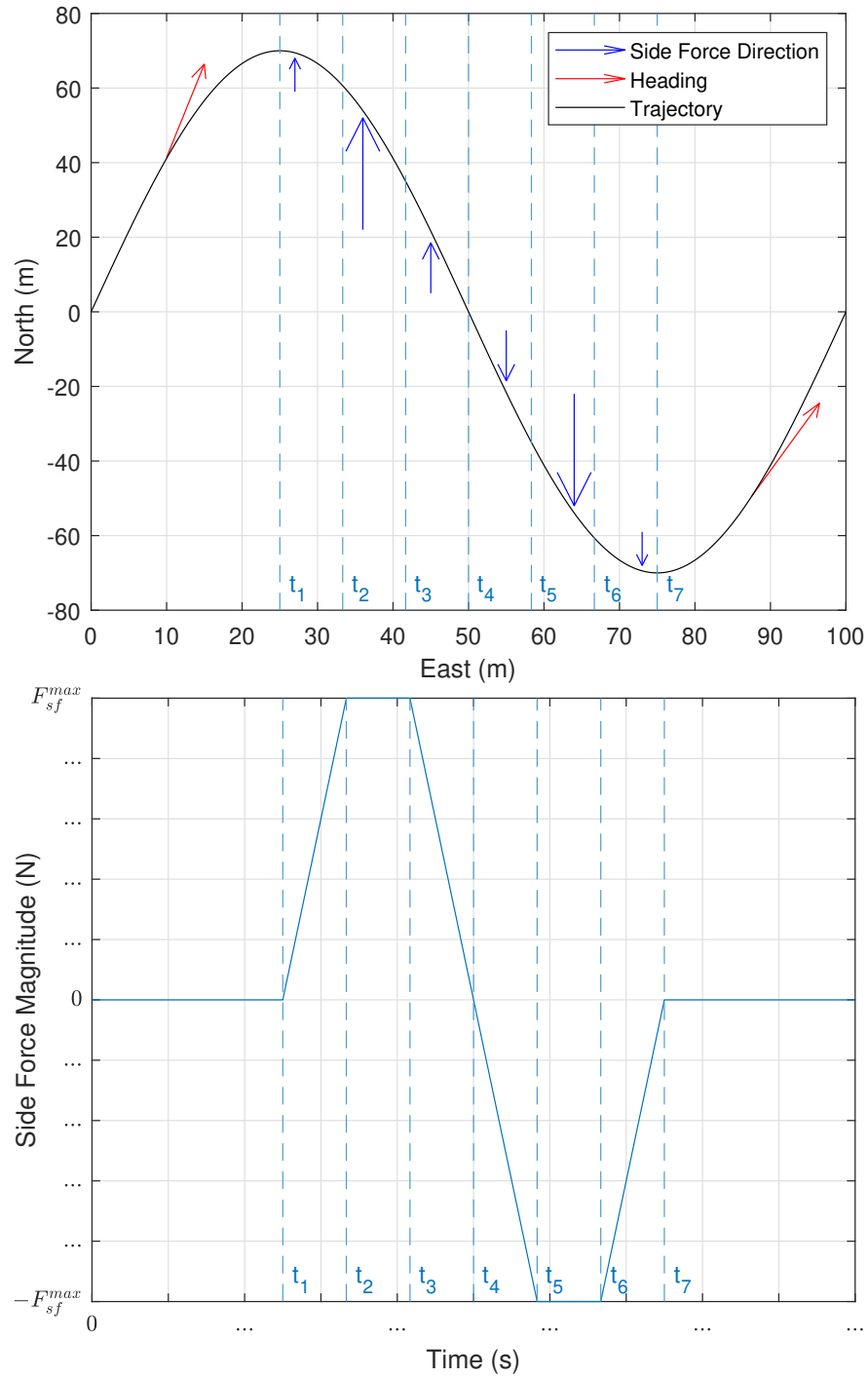


Figure 2.2: Sine-wave maneuver in the inertial frame (top) and side force magnitude profile (bottom).

Similar to \mathbf{p}_i^{ref} , ψ^{ref} was set to be constant at the end of the maneuver. Note that with the addition of yaw in this maneuver, more thrust will be demanded from the motors, further exacerbating actuator saturation.

Wind Disturbance Model

The main aspect of the maneuver that drives the vehicle further into saturation is a lateral disturbance force, intended to represent a wind disturbance. Fig. 2.2 illustrates how the direction and magnitude of the side force vary throughout the maneuver. The direction starts in the positive North direction from time t_1 to t_4 , then switches direction. The magnitude of this force increases and decreases linearly up to a maximum value of F_{sf}^{max} . The values for the timestamps t_1 to t_7 are listed in Table 2.1. The negative values for the side force magnitude are to simply indicate the change in direction. It's important to note that with this side force for large values of F_{sf}^{max} , the maneuver is now far beyond the actuator capabilities of the quadrotor and the control system seeks to execute this maneuver with as much tracking accuracy as possible. Therefore, any significant position tracking errors seen in simulations can largely be attributed to the quadrotor's actuator limitations rather than a flaw in the control system itself. This will be demonstrated in Sec. 2.2.4.

For the sake of better visualization of this side force, it is convenient to establish a correspondence between the side force magnitude and the wind speed. This can be achieved by using a frame drag model, as described in [38] and [39]:

$$F_{sf} = \frac{1}{2} \rho C_D A v_w^2 \quad (2.16)$$

$$v_w = \sqrt{\frac{2F_{sf}}{\rho C_D A}} \quad (2.17)$$

where v_w is the corresponding wind speed relative to the quadrotor for a given side force magnitude, F_{sf} . The air density is given by ρ , the drag coefficient by C_D , and the projected area "seen" by the wind by A . The 'flat plate area', $C_D A$, can be found experimentally by

Table 2.1: Side force timestamps

Parameter	Symbol	Value	Unit
Timestamp 1	t_1	7.333	s
Timestamp 2	t_2	8.346	s
Timestamp 3	t_3	9.360	s
Timestamp 4	t_4	10.370	s
Timestamp 5	t_5	11.383	s
Timestamp 6	t_6	12.397	s
Timestamp 7	t_7	13.407	s

measuring the drag force as the quadrotor flies through a known air flow [38]. In [40], a C_DA of 0.139 m^2 was found experimentally for the Spiri quadrotor platform, the platform that will be used for simulations in this thesis.

2.1.4 Simulation Parameters

The simulation parameters used for the aforementioned models are listed in Table 2.2. The vehicle parameters in this table were obtained from a Computer-Aided Design (CAD) model of the Spiri quadrotor. Its motor parameters were found via static thruster testing, as is done [34]. Note that in addition to minimum and maximum thrust constraints being imposed on the actuators, constraints are also imposed on the roll/pitch angles of the quadrotor (e.g. θ_{min}^d and θ_{max}^d). In general, constraints on these angles are optional and may be specified, for example, to accommodate an onboard camera that is constrained by the vehicle's tilt angle; or as a safety precaution against loss of level flight due to large roll/pitch angles. As previously mentioned, the acceleration a_E depends on vehicle characteristics like its mass, max thrust etc., listed in Table 2.2, such that there is sufficient saturation during the maneuver. The chosen acceleration is also defined in this table.

Table 2.2: Quadrotor, motor, and simulation parameters

Parameter	Symbol	Value	Unit
Mass	m	1.023	kg
Arm length	l	0.235	m
Moment of inertia	\mathbf{J}_b	$\begin{bmatrix} 0.0123 & 0 & 0 \\ 0 & 0.0127 & 0 \\ 0 & 0 & 0.0239 \end{bmatrix}$	$\text{kg} \cdot \text{m}^2$
Minimum desired pitch	θ_{min}^d	-60	deg
Maximum desired pitch	θ_{max}^d	60	deg
Minimum desired roll	ϕ_{min}^d	-60	deg
Maximum desired roll	ϕ_{max}^d	60	deg
Thrust constant	k_t	1.12×10^{-7}	N/RPM^2
Torque constant	k_q	$k_t/70$	$\text{N} \cdot \text{m}/\text{RPM}^2$
Maximum motor thrust	T_{max}	5.02	N
Minimum motor thrust	T_{min}	0	N
Motor inertia	J_r	2.01×10^{-5}	$\text{kg} \cdot \text{m}^2$
Rate constant	k_r	17	1/s
Thrust-to-Weight ratio	-	2	-
Flat plate area	$C_D A$	0.139	m^2
Eastern acceleration	a_E	0.93	m/s^2

2.2 Conventional Control System

In this section, we present a baseline conventional controller that has been widely used in the literature. As will be shown, it performs well without actuator saturation, but poorly when actuator saturation is reached.

2.2.1 Position Controller

Conventional position controllers for trajectory-tracking based on PD control can be found in [7] and [24]. These position controllers begin by computing a desired force vector using

a PD controller, as follows:

$$\mathbf{f}_i^{des} = \mathbf{K}_{pp}(\mathbf{p}_i^{ref} - \mathbf{p}_i) + \mathbf{K}_{pd}(\mathbf{v}_i^{ref} - \mathbf{v}_i) + mg \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (2.18)$$

where \mathbf{f}_i^{des} is the desired force vector expressed in the inertial frame (as denoted by subscript i) and \mathbf{v}_i is the velocity vector of the vehicle's center of mass. The inertial reference position and velocity vectors for tracking are given by \mathbf{p}_i^{ref} and \mathbf{v}_i^{ref} . The proportional and derivative gain matrices, \mathbf{K}_{pp} and $\mathbf{K}_{pd} \in \mathbb{R}^{3 \times 3}$, respectively, are diagonal matrices.

The desired attitude of the vehicle will be chosen so as to direct and modulate the vehicle's thrust vector (which is in the body z -direction) to match the desired force vector. That is:

$$\mathbf{f}_i^{des} = \mathbf{C}^T(\psi^{ref}, \theta^d, \phi^d) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} T_{total}^{des} \quad (2.19)$$

where $\mathbf{C}(\psi^{ref}, \theta^d, \phi^d)$ is the rotation matrix equivalent of the set of Euler angles defining the desired attitude. A yaw-pitch-roll (ZYX) Euler angle convention is used where ψ^{ref} , θ^d , and ϕ^d correspond to the reference/desired yaw, pitch, and roll angles, respectively. The superscript, ref , is used to denote that the yaw setpoints come from the motion planner, whereas the superscript, d , indicates that the pitch and roll setpoints come from the position controller, as will be discussed shortly.

The relationship between the rotation matrix, $\mathbf{C}(\psi, \theta, \phi)$, and the set of Euler angles (ZYX) is defined by:

$$\mathbf{C}(\psi, \theta, \phi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \quad (2.20)$$

where $c_\theta = \cos \theta$, $s_\theta = \sin \theta$, and this follows for the other angles as well. Substituting (2.20) into (2.19), we find:

$$\mathbf{f}_i^{des} = \begin{bmatrix} -(c_{\phi^d} s_{\theta^d} c_{\psi^{ref}} + s_{\phi^d} s_{\psi^{ref}}) \\ -(c_{\phi^d} s_{\theta^d} s_{\psi^{ref}} - s_{\phi^d} c_{\psi^{ref}}) \\ -(c_{\phi^d} c_{\theta^d}) \end{bmatrix} T_{total}^{des} \quad (2.21)$$

Considering a general trajectory where the vehicle's reference heading, ψ^{ref} , is defined throughout the maneuver (specified by the motion planner), (2.21) allows us to find analytical solutions for the desired attitude and total thrust [41], as follows:

$$T_{total}^{des} = \sqrt{(f_{i,x}^{des})^2 + (f_{i,y}^{des})^2 + (f_{i,z}^{des})^2} \quad (2.22)$$

$$\theta^d = \arctan \frac{\cos \psi^{ref} f_{i,x}^{des} + \sin \psi^{ref} f_{i,y}^{des}}{f_{i,z}^{des}} \quad (2.23)$$

$$\phi^d = \arcsin \frac{\cos \psi^{ref} f_{i,y}^{des} - \sin \psi^{ref} f_{i,x}^{des}}{T_{total}^{des}} \quad (2.24)$$

where $f_{i,x}^{des}$, $f_{i,y}^{des}$, and $f_{i,z}^{des}$ are the x , y , and z components of \mathbf{f}_i^{des} .

If minimum and maximum constraints are to be imposed on θ^d and ϕ^d , then a naive approach is to saturate their analytically computed values at these limits. When this is done, (2.21) is no longer satisfied, with the saturated values of θ^d and ϕ^d . This approach is categorized as 'conventional' in this thesis.

With $\mathbf{C}(\psi^{ref}, \theta^d, \phi^d)$ and T_{total}^{des} evaluated analytically, the controller computes a desired total thrust that considers the actual attitude of the vehicle, $\mathbf{C}(\mathbf{q})$. This is accomplished using the equation:

$$T_{total}^{body} = \mathbf{C}(\mathbf{q}) \left(\mathbf{C}^T(\psi^{ref}, \theta^d, \phi^d) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} T_{total}^{des} \right) \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (2.25)$$

where (\cdot) denotes the dot product and T_{total}^{body} is the new desired total thrust that will be sent to the mixer. This step projects the computed desired thrust vector onto the thrust axis of the current attitude.

The position controller is also responsible for outputting ω_b^d , the desired angular velocity vector expressed in the body frame. With the desired attitudes acquired from the aforementioned equations, (2.3) can be used to compute the desired angular velocity from the desired attitudes. However, it is possible to achieve controller convergence with a zero desired angular velocity vector, as shown in [19]. Thus, for simplicity, this approach was used for simulations.

2.2.2 Attitude Controller

Conventional attitude controllers such as the ones proposed in [7] and [17] are designed to control the full attitude error. The controller used in [7, 17] can be summarized in the following equations using quaternions:

$$\mathbf{q}^e = \mathbf{q}^* \odot \mathbf{q}^{des} \quad (2.26)$$

$$\omega_b^e = \omega_b^d - \omega_b \quad (2.27)$$

$$\mathbf{e}_b = 2q_0^e \mathbf{q}_{1:3}^e \quad (2.28)$$

$$\mathbf{m}_b^{des} = \mathbf{J}_b (\mathbf{K}_{ap} \mathbf{e}_b + \mathbf{K}_{ad} \omega_b^e) \quad (2.29)$$

where \mathbf{q}^* is the inverse of the quaternion defining the attitude of the vehicle. The error quaternion is denoted by \mathbf{q}^e , and its scalar and vector components by q_0^e and $\mathbf{q}_{1:3}^e$, respectively. The proportional and derivative gain matrices, \mathbf{K}_{ap} and $\mathbf{K}_{ad} \in \mathbb{R}^{3 \times 3}$, respectively, are diagonal matrices for this controller. The desired moment vector resolved in the body frame, \mathbf{m}_b^{des} , thus computed by the controller is sent to the mixer. The desired attitude is defined by the unit quaternion \mathbf{q}^{des} , which can be computed using the set of Euler angles obtained from the position controller.

A unit quaternion equivalent to the set of Euler angles (ZYX) can be computed using the following equation:

$$\mathbf{q}(\psi, \theta, \phi) = \begin{bmatrix} c_{\psi/2}c_{\theta/2}c_{\phi/2} + s_{\psi/2}s_{\theta/2}s_{\phi/2} \\ c_{\psi/2}c_{\theta/2}s_{\phi/2} - s_{\psi/2}s_{\theta/2}c_{\phi/2} \\ c_{\psi/2}s_{\theta/2}c_{\phi/2} + s_{\psi/2}c_{\theta/2}s_{\phi/2} \\ s_{\psi/2}c_{\theta/2}c_{\phi/2} - c_{\psi/2}s_{\theta/2}s_{\phi/2} \end{bmatrix} \quad (2.30)$$

where $c_{\theta/2} = \cos \frac{\theta}{2}$, $s_{\theta/2} = \sin \frac{\theta}{2}$, and analogously for the other angles.

2.2.3 Mixer

The mixer is responsible for distributing the control efforts among the actuators. Using (2.7) and (2.9), the control allocation problem can be formulated as:

$$\mathbf{\Gamma} \mathbf{x} = \mathbf{b}$$

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{k_q}{k_t} & \frac{k_q}{k_t} & \frac{-k_q}{k_t} & \frac{-k_q}{k_t} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} T_{total}^{body} \\ \mathbf{m}_b^{des} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} T_1^{des} \\ T_2^{des} \\ T_3^{des} \\ T_4^{des} \end{bmatrix} \quad (2.31)$$

where T_i^{des} is the desired thrust from the i^{th} motor. The conventional method for solving this control allocation problem, as described in [4, 10, 12], is to compute the exact solution analytically, as:

$$\mathbf{x} = \mathbf{\Gamma}^{-1} \mathbf{b} \quad (2.32)$$

If the computed desired thrust values violate the minimum and maximum thrust constraints (T_{min} and T_{max}), i.e. there is actuator saturation, then these values are simply saturated at the constraint values.

Table 2.3: Controller gains and disturbance characteristics

Gain	Symbol	Value	Unit
Maximum side force magnitude	F_{sf}^{max}	$mg/2$	N
Maximum wind speed	v_w^{max}	7.52	m/s
Position Proportional	\mathbf{K}_{pp}	$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	N/m
Position Derivative	\mathbf{K}_{pd}	$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$	N · s/m
Attitude Proportional	\mathbf{K}_{ap}	$\begin{bmatrix} 120 & 0 & 0 \\ 0 & 120 & 0 \\ 0 & 0 & 120 \end{bmatrix}$	1/s ²
Attitude Derivative	\mathbf{K}_{ad}	$\begin{bmatrix} 16 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 16 \end{bmatrix}$	1/s

2.2.4 Simulations with Conventional Control System

The simulation environment was created in Simulink/MATLAB. Aerodynamic drag was not included in the simulation. Similarly, no model for a state estimator was used, and so the control system assumed perfect state feedback from its sensors. As previously mentioned, maximum and minimum roll/pitch constraints are considered in addition to actuator (thrust) constraints, with their values listed in Table 2.2.

Without Actuator Constraints

The purpose of the first simulation was to illustrate the overall effectiveness of the conventional control scheme without the presence of actuator saturation. Therefore, no actuator constraints were considered; the thrusters were assumed to be capable of generating whatever thrust was demanded. The control gains and side force magnitude (with its corresponding wind speed) used in this simulation are listed in Table 2.3. The wind speed shown in this table would be considered quite high for a quadrotor of this size.

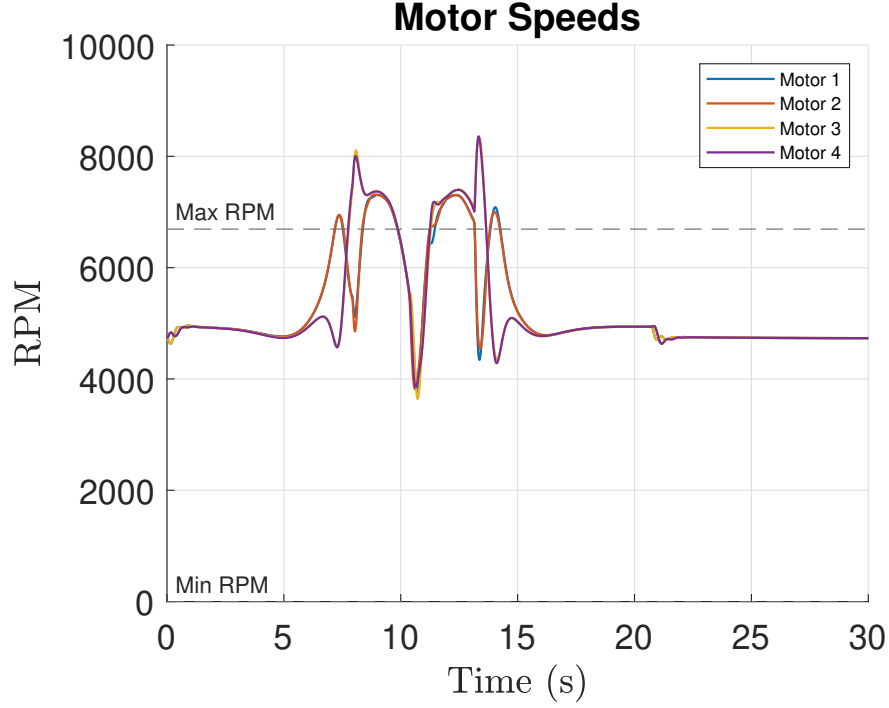


Figure 2.3: Motor speeds of all four motors vs. Time, using the conventional approach without actuator constraints, $F_{sf}^{max} = mg/2$ corresponding to wind speed = 7.52 m/s .

The motor speeds of all of four motors are plotted in Fig. 2.3 This figure shows us that the maneuver, alongside the wind disturbance, is aggressive enough that the vehicle would require more thrust than the motors are capable of providing. This is exemplified by the maximum RPM value noted in the figure, which corresponds to the maximum thrust and thrust constant shown in Table 2.2. The peak thrust reached in this maneuver is roughly 50% more than the maximum thrust in Table 2.2. Fig. 2.4 illustrates the position and attitude tracking performance in this scenario. The discrepancies between reference and actual values of the inertial position are mostly attributable to the saturation of the pitch angle which first starts $t = 8s$. As noted earlier, when this occurs the thrust vector is no longer pointed in the desired direction, leading to position errors in all three directions. Nonetheless, the performance shown in Fig. 2.4 can be considered reasonable and the vehicle remains stable throughout the maneuver.

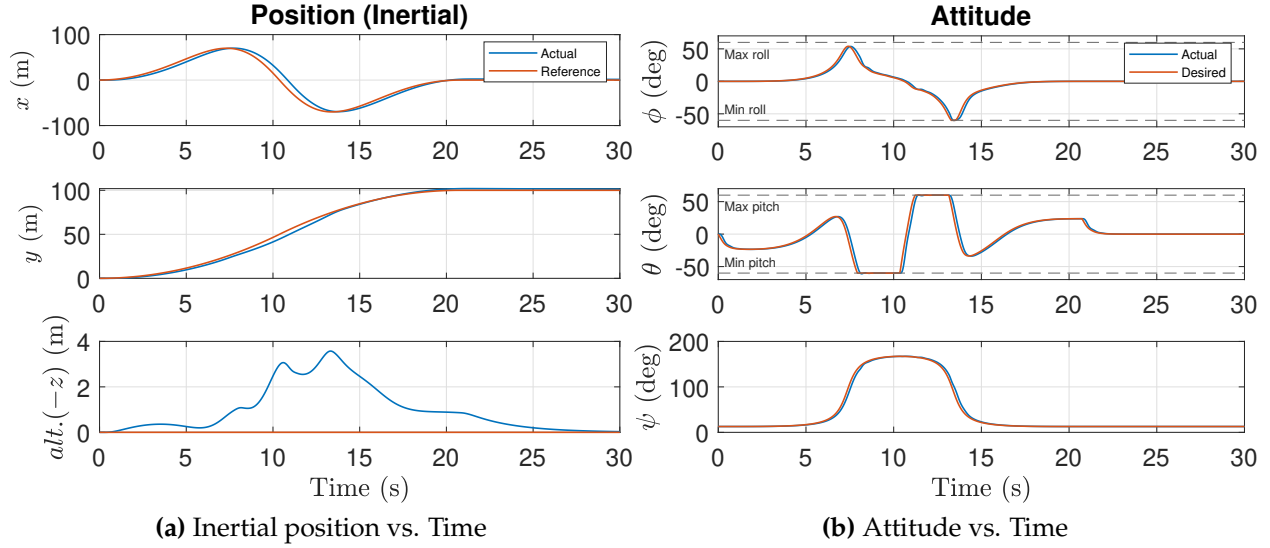


Figure 2.4: Simulation plots using: the conventional approach without actuator constraints, wind speed = 7.52 m/s .

With Actuator Constraints

The same simulation was conducted but with the actuator constraints now imposed. The motor speeds, position, and attitude data of this simulation are plotted in Fig. 2.5 and 2.6, respectively. It is now obvious that by imposing actuator constraints, the conventional control system is unable to intelligently handle this scenario and the vehicle becomes unstable. These findings motivate this thesis into investigating potential modifications that can be made to the conventional control system to mitigate these adverse effects from actuator saturation.

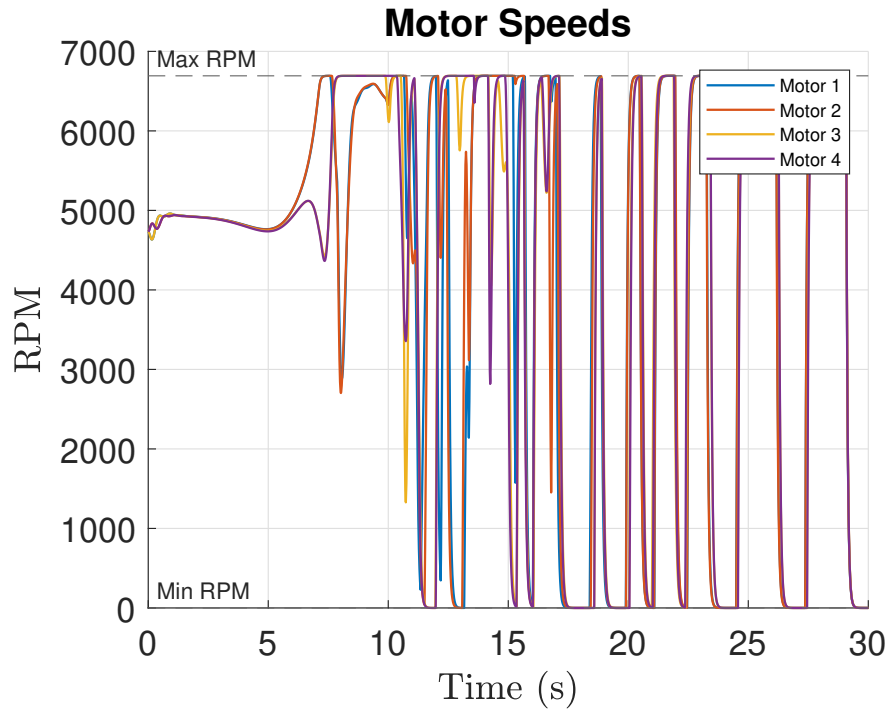


Figure 2.5: Motor speeds of all four motors vs. Time, using the conventional approach with actuator constraints imposed, wind speed = 7.52 m/s .

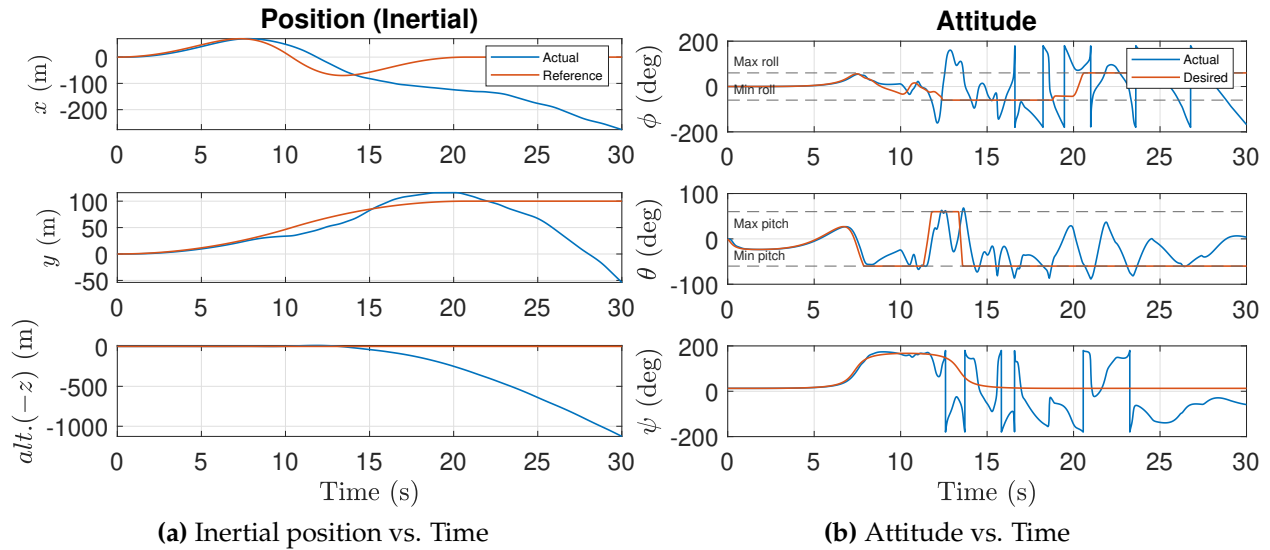


Figure 2.6: Simulation plots using: the conventional approach with actuator constraints imposed, wind speed = 7.52 m/s .

Chapter 3

Control System Improvements

In this chapter, we discuss the modifications made to the position controller, attitude control, and mixer. The motivation behind each modification will be highlighted, starting from the lowest level of the cascaded structure (the mixer) to the highest level (the position controller).

3.1 Optimal Mixer

As previously discussed in Sec. 1.1.1, the conventional mixer suffers from degraded thrust, roll, pitch, and yaw control because of thrust clipping normally used to handle actuator saturation. This approach ignores how each control effort contributes to the stability and performance of the vehicle [4, 10], and so the vehicle experiences degraded stability and performance in the event of saturation, as was evident in Sec. 2.2.4. To address this issue, a more sophisticated approach to thrust mixing is proposed in [6, 12, 14] where the control allocation problem is formulated as a constrained weighted least squares prob-

lem, as follows:

$$\begin{aligned} \min : J &= \|\mathbf{W}_m(\mathbf{b} - \mathbf{\Gamma}\mathbf{x})\|^2 \\ \text{s.t. : } \begin{bmatrix} T_{min} \\ T_{min} \\ T_{min} \\ T_{min} \end{bmatrix} &\leq \begin{bmatrix} T_1^{des} \\ T_2^{des} \\ T_3^{des} \\ T_4^{des} \end{bmatrix} \leq \begin{bmatrix} T_{max} \\ T_{max} \\ T_{max} \\ T_{max} \end{bmatrix} \end{aligned} \quad (3.1)$$

where:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} & \frac{\sqrt{2}l}{2} & \frac{-\sqrt{2}l}{2} \\ \frac{k_q}{k_t} & \frac{k_q}{k_t} & \frac{-k_q}{k_t} & \frac{-k_q}{k_t} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} T_{total}^{body} \\ \mathbf{m}_b^{des} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} T_1^{des} \\ T_2^{des} \\ T_3^{des} \\ T_4^{des} \end{bmatrix} \quad (3.2)$$

The definitions of $\mathbf{\Gamma}$, \mathbf{b} , and \mathbf{x} are repeated from (2.31) for convenience. The symbol, $\|\cdot\|$, denotes the Euclidean norm of the vector in (\cdot) . This optimization problem effectively tries to select actuator thrusts in \mathbf{x} that come ‘as close as possible’ to meeting the desired body forces and moments in \mathbf{b} , while ensuring that the thrusts stay within their feasible bounds. The weighting matrix, $\mathbf{W}_m \in \mathbb{R}^{4 \times 4}$, is a diagonal positive definite matrix, in which the entries define the relative prioritization between the control efforts T_{total}^{body} , $m_{b,x}^{des}$, $m_{b,y}^{des}$, and $m_{b,z}^{des}$, found in \mathbf{b} .

This type of mixer is termed the ‘optimal mixer’ in this thesis, and is an improvement over the conventional approach since we are now able to prioritize certain control efforts over others within the actuator constraints. For example, we can choose to prioritize the control efforts $m_{b,x}^{des}$ and $m_{b,y}^{des}$ over the others in order to improve the vehicle’s roll and pitch control. This would result in improved stability and performance of the vehicle since the reduced attitude (combination of roll and pitch) control is largely responsible for the vehicle’s stability and position control [4,6,10]. Fortunately, the optimization problem in (3.1) can be formulated as a quadratic program (QP), for which numerous efficient solvers exist for real-time applications, such as those found in [6,14,15].

To demonstrate the capabilities of the optimal mixer, we can perform the same simulation as in Sec. 2.2.4 but with the conventional mixer replaced with the optimal mixer.

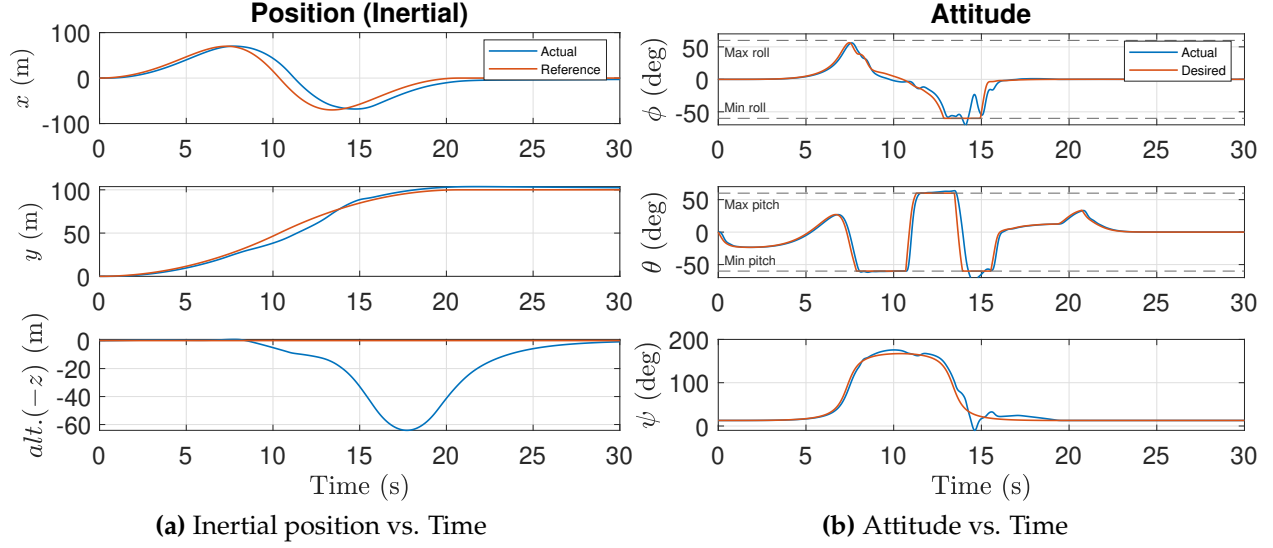


Figure 3.1: Simulation plots using: conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{1, 1000, 1000, 100\}$), side force magnitude = $mg/2$, wind speed = 7.52 m/s .

The weighting matrix for this simulation is set to:

$$\mathbf{W}_m = \begin{bmatrix} 1N^{-1} & 0 & 0 & 0 \\ 0 & 1000N^{-1}m^{-1} & 0 & 0 \\ 0 & 0 & 1000N^{-1}m^{-1} & 0 \\ 0 & 0 & 0 & 100N^{-1}m^{-1} \end{bmatrix} = \text{diag}\{1, 1000, 1000, 100\} \quad (3.3)$$

This weighting matrix forces the optimal mixer to prioritize the control efforts for roll and pitch control first, $m_{b,x}^{des}$ and $m_{b,y}^{des}$ (weight of 1000 each), then the control effort for yaw control, $m_{b,z}^{des}$ (weight of 100), and finally the total thrust, T_{total}^{body} (weight of 1). The magnitudes of these weights were chosen based on open-loop testing of the optimal mixer where the weights were varied until the desired level of prioritization was achieved. Details of the open-loop testing are discussed in Appendix A. Furthermore, it should be noted that the units in this matrix are not consistent; one weight is associated with a force and three weights are associated with moments, and this is also discussed in Appendix A.

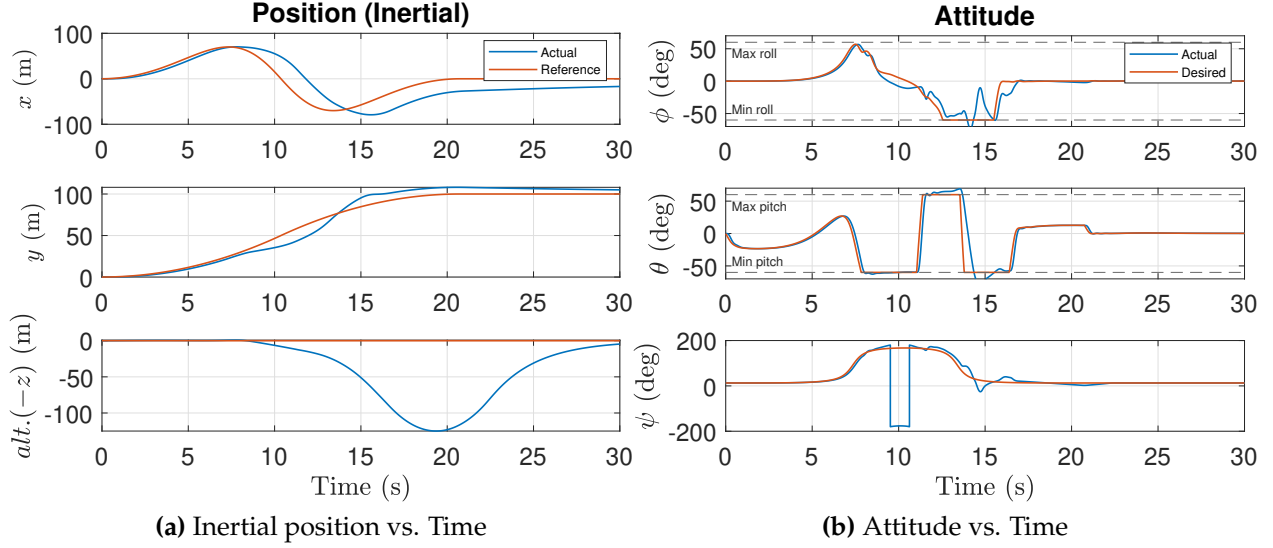


Figure 3.2: Simulation plots using conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{1, 1000, 1000, 100\}$), side force magnitude = mg , wind speed = 10.6m/s .

Position and attitude data from this simulation are plotted in Fig. 3.1. In this figure, we can see that the vehicle is now capable of executing the maneuver without becoming unstable, unlike with the conventional mixer (results shown in Fig. 2.6). We can attribute this improvement to the enhanced reduced attitude control granted by the optimal mixer, as is clear in the reasonable attitude tracking shown in Fig. 3.1b. However, there is significantly more degradation in the position control than the case where the conventional mixer was used with no actuator constraints (shown in Fig. 2.4). There are three causes for this: the first is, quite obviously, because actuator constraints are now being considered in the simulation, and so the vehicle has limited thrust under a severe wind disturbance. Second, as noted for the case in Sec. 2.2.4, starting at $t = 8\text{s}$ there is saturation of the pitch and roll angles which prevents the thrust vector from being pointed in the desired direction, further exacerbating the degradation in position control. Finally, we know that the vehicle's thrust vector is indifferent to the vehicle yaw and that the total thrust is involved in position control, as shown in (2.1). The weighting matrix selected chooses to prioritize yaw control ($m_{b,z}^{des}$) over total thrust control (T_{total}^{body}), and so we get diminished position control in exchange for better yaw control.

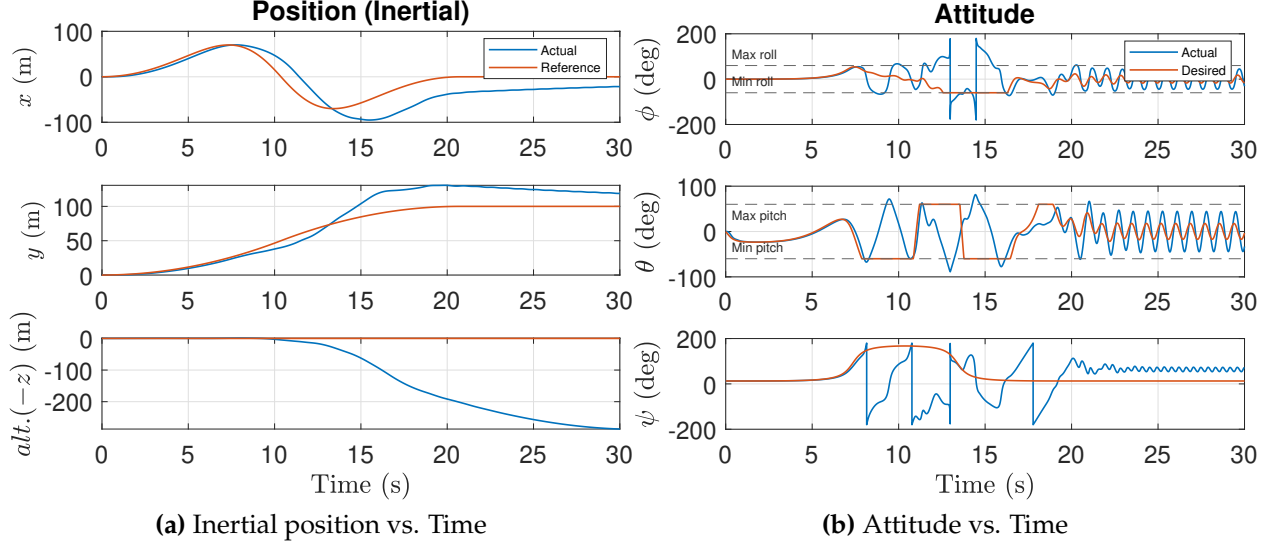


Figure 3.3: Simulation plots using conventional position and attitude controllers, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

It is imperative that the position control performance be improved for any practical application of the optimal mixer. This is apparent from the altitude control performance, where a near 60 m drop in altitude, shown in Fig. 3.1a, would likely lead to the vehicle crashing into the ground, and this drop would worsen under more severe wind conditions. To illustrate this, we can double the side force magnitude and run the same simulation, the results of which are plotted in Fig. 3.2. With a side force magnitude of mg (corresponding to a wind speed of 10.6 m/s), the altitude drop is doubled. The position control in the North (x) and East (y) directions are also worsened due to the longer saturation times of the roll and pitch angles. However, the quadrotor still remains stable.

Of the three aforementioned causes of diminished position control, only the third can be mitigated in the optimal mixer by choosing to prioritize total thrust control over yaw control. We can formulate a new weighting matrix, as $\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$. These weights were obtained using the same open-loop mixer tests described in Appendix A. With these weights, we are relinquishing yaw control in exchange for improved roll, pitch, and thrust control in the event of actuator saturation. It should be noted that, according to (2.1), the position control is dictated by the vehicle's thrust vector, whose

direction and magnitude is dictated by the reduced attitude and total thrust. However, with this new weighting matrix, we are stating that the reduced attitude control should be given priority over the total thrust control. This follows the reasoning shared in [6]: The total thrust can only be applied in the right direction if the vehicle has the right attitude. Additionally, the vehicle's attitude dynamics are faster than its translational dynamics, meaning that the attitude control objectives can be achieved faster than the position control ones.

Using this new weighting matrix in the same simulation conditions described in Fig. 3.2 we find that the vehicle becomes unstable, as shown in Fig. 3.3. This is an unexpected result since, as previously mentioned, the vehicle's thrust vector is indifferent to the vehicle yaw and so it was expected that relinquishing yaw control would have no effect on the vehicle's overall stability. When first confronted with this phenomenon, it was assumed that perhaps the weights or control gains had not been tuned correctly, but re-tuning of these values would not correct the issue. After further investigation, it was found that the instability was a natural consequence of the conventional attitude controller being used alongside this new weighting matrix.

3.2 Decoupled Attitude Controller

The conventional attitude controller is designed to control the full attitude error, as shown in (2.29). Consequently, the reduced attitude error dynamics of the vehicle are coupled with the yaw error dynamics [10,20]. As a result, any lingering yaw error indirectly causes a degradation in reduced attitude control, thereby causing the vehicle to lose performance or become unstable. In the last simulation (shown in Fig. 3.3), the weighting matrix was selected to relinquish yaw control, and the resultant lingering yaw error indirectly caused the vehicle to become unstable.

In [10], this issue is addressed by proposing a decoupled attitude controller to complement their heuristic formulation of an optimal mixer; the authors propose splitting and

replacing the full attitude error with its reduced and yaw attitude components, with gains allocated separately for each component. In doing so, they were able to effectively decouple the reduced and yaw attitude error dynamics. Their controller can be summarized in the following equations after finding \mathbf{q}^e from (2.26):

$$\mathbf{q}^e = \mathbf{q}^{e,red} \odot \mathbf{q}^{e,yaw} \quad (3.4)$$

$$\mathbf{q}^{e,red} = \frac{1}{\sqrt{q_0^{e^2} + q_3^{e^2}}} \begin{bmatrix} q_0^{e^2} + q_3^{e^2} \\ q_0^e q_1^e - q_2^e q_3^e \\ q_0^e q_2^e + q_1^e q_3^e \\ 0 \end{bmatrix} \quad (3.5)$$

$$\mathbf{q}^{e,yaw} = \frac{1}{\sqrt{q_0^{e^2} + q_3^{e^2}}} \begin{bmatrix} q_0^e \\ 0 \\ 0 \\ q_3^e \end{bmatrix} \quad (3.6)$$

$$\mathbf{m}_b^{des} = \mathbf{J}_b \left(k_{red} \mathbf{q}_{1:3}^{e,red} + k_{yaw} \text{sgn}(q_0^e) \mathbf{q}_{1:3}^{e,yaw} + \mathbf{K}_{ad} \boldsymbol{\omega}_b^e \right) \quad (3.7)$$

where $\mathbf{q}^{e,red}$ and $\mathbf{q}^{e,yaw}$ are the reduced and yaw attitude errors in quaternion form. The first, second, and third components of $\mathbf{q}_{1:3}^e$ are denoted by q_1^e , q_2^e , and q_3^e . The gains for the reduced and yaw attitude control, k_{red} and k_{yaw} , are scalar values.

It is noteworthy that, in addition to decoupling the error dynamics, this new controller enables us to set individual gains for the reduced and yaw attitude control. This is in contrast to the conventional attitude controller described in (2.29) where, due to the coupling, one would incorrectly assume that the three diagonal entries in \mathbf{K}_{ap} correspond to independent gains for roll, pitch, and yaw control.

Since we have selected a weighting matrix that aims to relinquish yaw control, it would make sense to consider setting k_{yaw} to zero in order to ‘assist’ the mixer in achieving this. However, the goal of the mixer is to only relinquish yaw control during periods of actuator saturation, and so k_{yaw} should be non-zero so that the vehicle may track the given yaw trajectory when there is no saturation. This is the main reason why decoupled

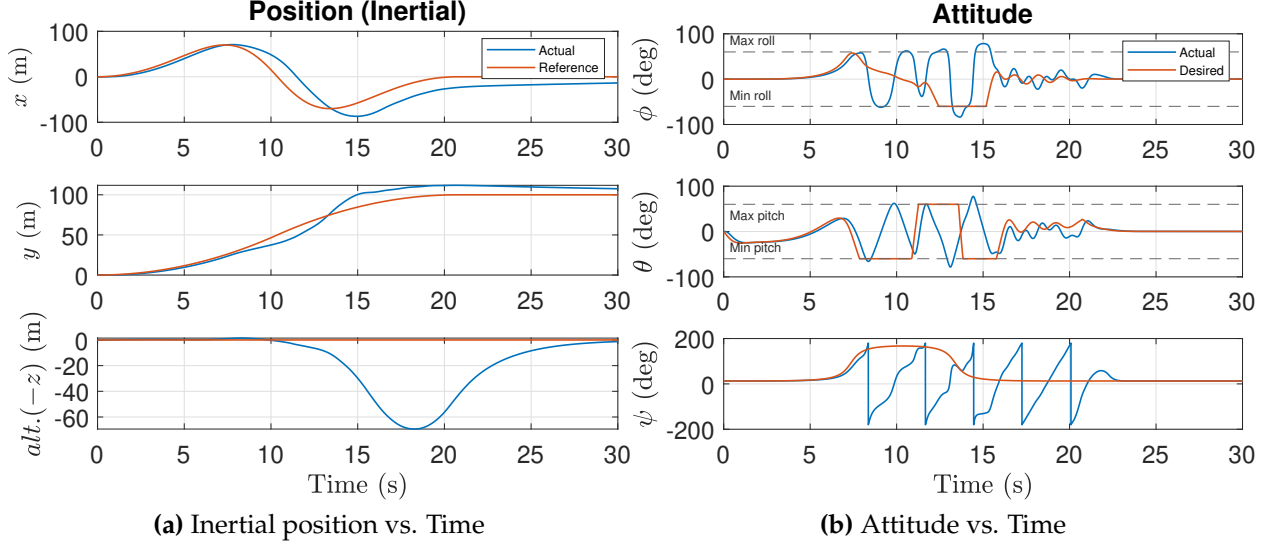


Figure 3.4: Simulation plots using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

attitude controllers similar to those proposed in [5] were not used in our work, as the decoupling properties were only attainable with a near-zero weight on yaw motion.

Another simulation was conducted after replacing the conventional attitude controller by the decoupled controller using a gain of 120 1/s^2 and 16 1/s^2 for k_{red} and k_{yaw} , respectively, alongside the optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$). The results of this simulation are shown in Fig. 3.4, which shows that the vehicle can now execute the maneuver without becoming unstable. Furthermore, we find that using the new weighting matrix improves the altitude control substantially relative to the case where the yaw control was prioritized over the total thrust, shown in Fig. 3.2: the maximum drop in altitude is around 70 m compared to 120 m previously. As for the North and East position control performance, we see relatively lower tracking errors in both directions between 10 and 13 seconds, but slightly worse afterwards.

In Fig. 3.4b, it's worth noting that there is a sizable difference between the actual and desired attitudes. This stems from the fact that, as stated previously, the optimal mixer prioritizes the control of the reduced attitude error instead of the full attitude error during

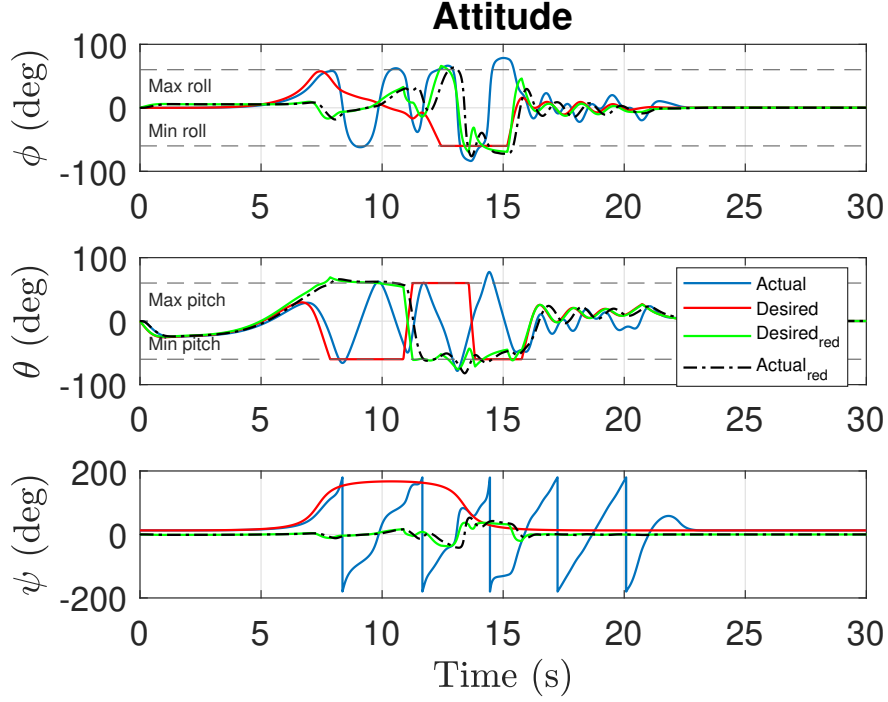


Figure 3.5: The reduced attitude components of the desired and actual attitudes in Fig. 3.4b.

actuator saturation. The reduced attitude components of the desired and actual attitudes are plotted in Fig. 3.5, and it can be concluded from these plots that there is accurate tracking of the desired attitude's reduced component throughout the maneuver.

The decoupled attitude controller has helped us in achieving overall stability of the vehicle, but the altitude drop is still excessive. It would make sense to readjust the weighting matrix of the optimal mixer to see whether altitude control could be further improved in exchange for poorer yaw control. We evaluated this by running another simulation with an increase in the weight corresponding to the desired total thrust, such that the new weighting matrix becomes: $\mathbf{W}_m = \text{diag}\{20, 1000, 1000, 1\}$. The results of this simulation are plotted in Fig. 3.6. Comparing the plots in this figure to those in Fig. 3.4 and Fig. 3.5, we see that the altitude control has actually become slightly worse: the maximum drop in altitude increases from around 65 m to 80 m. We can make two observations from these results: (1) the maximum amount of yaw control that could be relinquished was already achieved by the matrix: $\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$, and (2) the roll and pitch

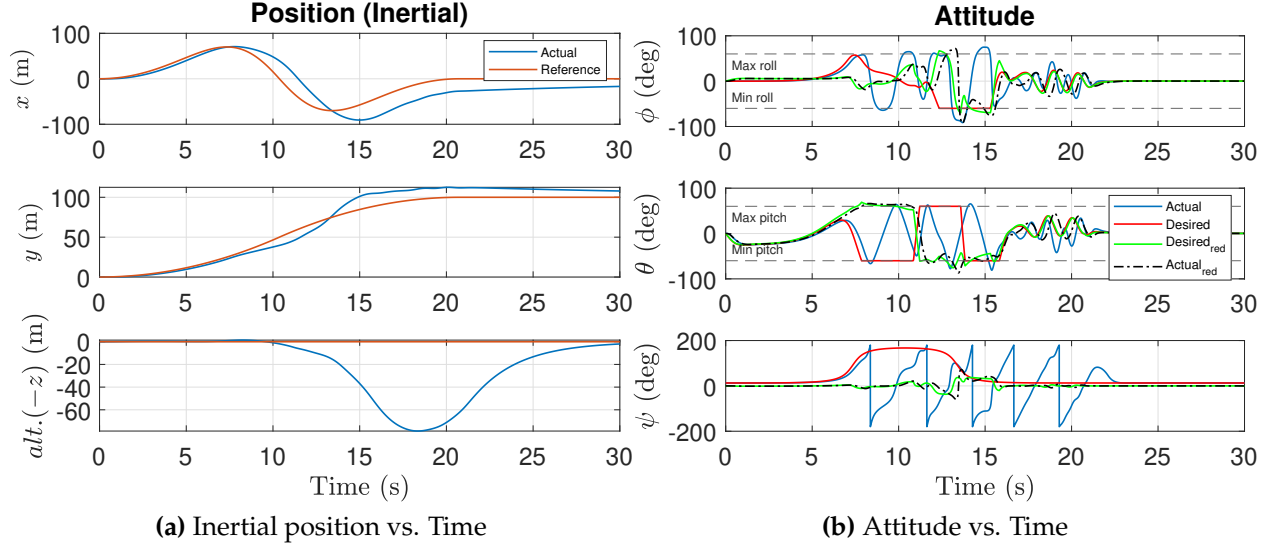


Figure 3.6: Simulation plots using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{20, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

control is being diminished in order to improve the total thrust control with the matrix: $\mathbf{W}_m = \text{diag}\{20, 1000, 1000, 1\}$. The first observation comes from the fact that there is virtually no difference in the yaw control in the two cases. The second observation can be seen in Fig. 3.6b where, towards the end of the maneuver ($t = 15\text{s}$ to $t = 23\text{s}$), the roll and pitch control is noticeably more erratic than in Fig. 3.5. It was found that any further increases in the weight corresponding to the desired total thrust led to further degradation in the roll and pitch control.

Recall that in Sec. 3.1, we claimed that the reduced attitude control should be given priority over the total thrust control based on the hypothesis that improved reduced attitude control leads to better overall position control. The second observation made from the above results gives weight to this claim, as we see diminished altitude control as a consequence of diminished reduced attitude control. Additional simulations were conducted where the weights corresponding to roll and pitch control were adjusted instead, and it was found that no better altitude control could be achieved than that seen in Fig. 3.4.

Overall, the enhanced thrust control (relative to the case seen in Fig. 3.2) seems to have improved the position control considering the substantial improvement in the altitude control. Nevertheless, given the excessive altitude drop, we must consider changes to the position controller to further mitigate this altitude loss. However, before doing this, we'll first consider a possible improvement to the attitude control that is rendered possible by the decoupled attitude controller.

3.2.1 Minimum Yaw Control

Before introducing the changes to the position controller, it's worth investigating the use of a minimum yaw constraint in the optimal mixer. This feature has now been made possible due to the decoupling of the yaw control by the decoupled attitude controller. This decoupling ensures us that the minimum yaw constraint will only affect the yaw control, and will not have an indirect impact on the reduced attitude control.

In Sec. 3.1, we designed the optimal mixer's weighting matrix to relinquish yaw control in exchange for improved tracking of the other control efforts. However, in certain applications, there may be a need for some level of control on the quadrotor's heading. Such applications may include the use of an onboard camera for area coverage or surveillance where guaranteed heading control of the camera is imperative. Furthermore, there may be unintended consequences associated with uncontrollable yaw in real-life scenarios such as, for example, erratic behaviour of the onboard state estimator. In [4], the authors propose a heuristic formulation of the optimal mixer where the user can, optionally, allocate some minimum percentage of yaw control. To achieve this, the authors force their mixer to apply a minimum percentage of $m_{b,z}^{des}$, the control effort associated with yaw control. This feature can be replicated in the optimal mixer by adding an additional

inequality constraint to the optimization problem in (3.1), as follows:

$$\begin{aligned}
\min : J &= \|\mathbf{W}_m(\mathbf{b} - \mathbf{\Gamma}\mathbf{x})\|^2 \\
s.t. : & \begin{bmatrix} T_{min} \\ T_{min} \\ T_{min} \\ T_{min} \end{bmatrix} \leq \begin{bmatrix} T_1^{des} \\ T_2^{des} \\ T_3^{des} \\ T_4^{des} \end{bmatrix} \leq \begin{bmatrix} T_{max} \\ T_{max} \\ T_{max} \\ T_{max} \end{bmatrix} \\
& \left| \frac{k_q}{k_t} T_1^{des} + \frac{k_q}{k_t} T_2^{des} - \frac{k_q}{k_t} T_3^{des} - \frac{k_q}{k_t} T_4^{des} \right| \geq \text{saturate} \left(|h * m_{b,z}^{des}|, 2 \frac{k_q}{k_t} (T_{max} - T_{min}) \right)
\end{aligned} \tag{3.8}$$

where:

$$\text{saturate}(x, y) = \begin{cases} x & \text{for } x \leq y \\ y & \text{for } x \geq y \end{cases} \tag{3.9}$$

The symbol, $|\cdot|$, gives the absolute value of the number in (\cdot) . The minimum percentage of yaw control is given by h as a fraction (e.g., 10% minimum yaw control means $h = 0.1$). To clarify the new equality introduced in (3.8): its left-hand side represents the yaw moment that will be generated about the body z-axis (obtained from the last row in (2.9)). We want this quantity to be at least equal to some percentage of the desired yaw moment, $h * m_{b,z}^{des}$. However, we must avoid excessive requests in cases where the desired yaw moment is beyond the thruster capabilities, and this leads to the introduction of the saturate function on the right-hand side. That maximum capability is $2 \frac{k_q}{k_t} (T_{max} - T_{min})$, and is found by using the equation on the left-hand side and setting T_1^{des} & T_2^{des} as T_{max} , and T_3^{des} & T_4^{des} as T_{min} .

Use of a minimum yaw constraint as an inequality constraint in the optimization ensures that the optimal mixer enforces this constraint at every point in the maneuver. One may also attempt to achieve this objective by simply modifying the corresponding weight on $m_{b,z}^{des}$ in the weighting matrix and solving the original optimization problem in (3.1). However, given that the optimization is solved numerically, there is no analytical approach for finding the precise weight needed for a given h . Therefore, in this scenario,

there is no guarantee that the optimal mixer will satisfy the minimum yaw constraint at every point, for all maneuvers.

To demonstrate the capabilities of this new formulation of the optimal mixer, we conducted the same simulation as in Fig. 3.4 but with a minimum 10% yaw control constraint applied to the optimal mixer, with plots shown in Fig. 3.7 (simulation results from Fig. 3.4 are also repeated for convenience). From this figure, we find that there is certainly an improvement in the yaw control with the minimum yaw constraint. Moreover, there is a degradation in position control, since we see larger tracking errors in the North, East, and altitude directions. This degradation is due to the fact that there is now some priority being given to yaw control, which takes some motor thrust away from tracking the other control efforts: T_{total}^{body} , $m_{b,x}^{des}$, and $m_{b,y}^{des}$.

In addition to a minimum yaw constraint, it is also possible to include rate constraints in the optimal mixer, as is done in [35]. Rate constraints involve restricting the rate at which thrust can vary according to the physical limitations of the motors, so that the output of the mixer may be more realistic. The use of rate constraints is investigated in Appendix B and it is concluded that they would not be beneficial in the present work.

3.3 Optimal Position Controller

In Sec. 3.1, it was noted that one of the sources of degraded position control was the saturation of the roll and pitch angles, as is done by the conventional position controller, preventing the thrust vector from being pointed in the desired direction (the desired thrust vector being \mathbf{f}_i^{des} in (2.18)). This phenomenon is similar to that found with the conventional mixer, where the naive saturation technique of the thrust values (thrust clipping) obtained from (2.32) resulted in a control effort vector that was different from the desired (\mathbf{b} in (2.31)). Given the similarities with the optimal mixer, it would therefore make sense to propose a constrained control allocation approach for position control to handle actuator and attitude constraints more intelligently.

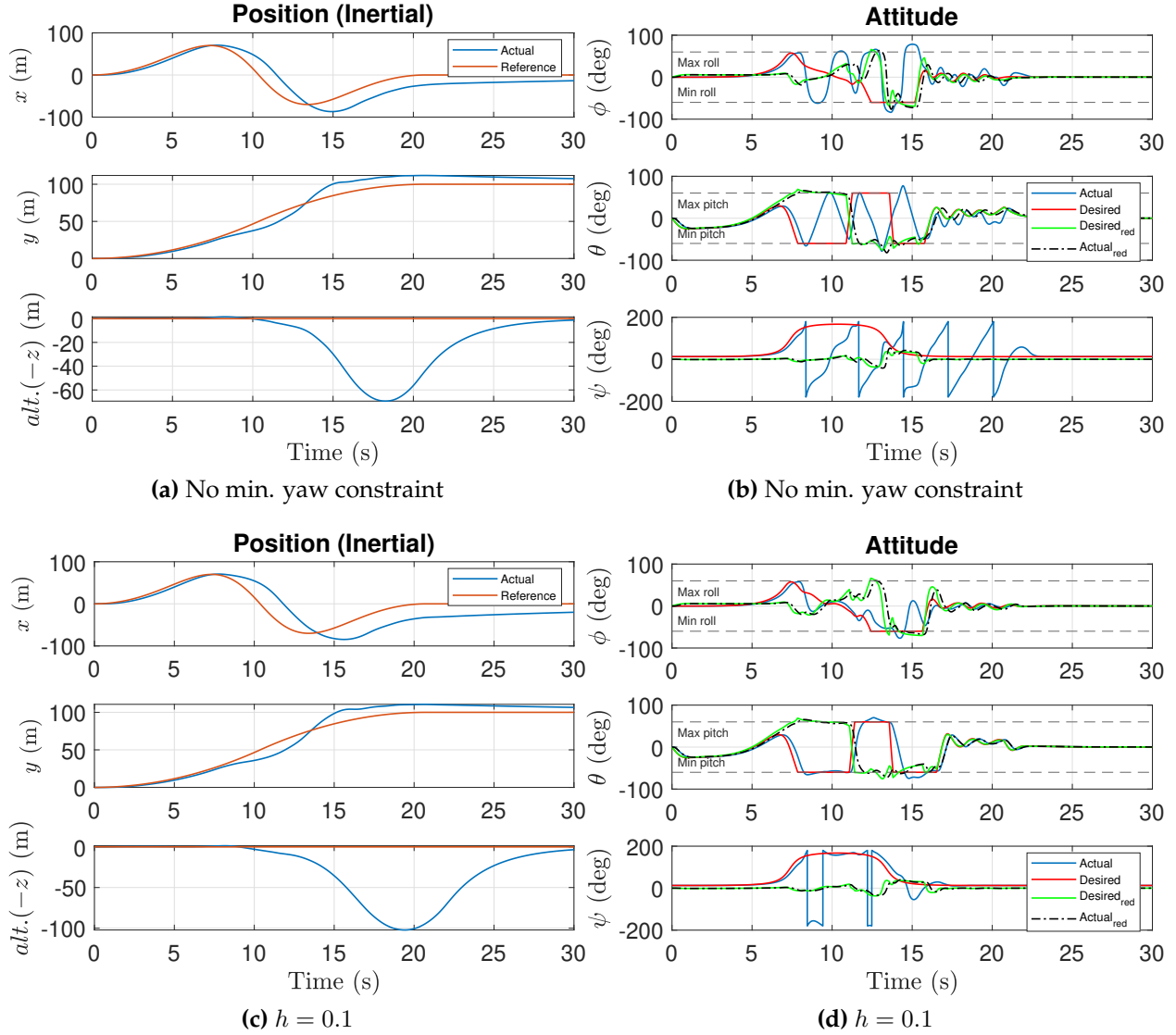


Figure 3.7: Simulation plots with (bottom) and without (top) minimum yaw constraint using: conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

In [25], the authors investigate precisely this approach for position control: Equation (2.21) is used to formulate a constrained least squares optimization problem in which the desired attitude $(\psi^d, \theta^d, \phi^d)$, and total thrust, T_{total}^{des} , are decision variables. However, their formulation of the optimization problem does not include a means to prioritize particular control actions. In this thesis, we seek to expand this control allocation approach by incorporating a weighting matrix to allow for relative control effort prioritization, similar to the optimal mixer. In doing so, we can choose to first prioritize the control effort corresponding to altitude control, $f_{i,z}^{des}$, to further mitigate the loss in altitude (seen previously) and reducing the vehicle's risk of crashing into the ground.

The proposed position controller follows the same steps as its conventional counterpart, from (2.18) to (2.21). To solve for the desired total thrust and attitude from (2.21), given constraints on the total thrust and attitude, the authors in [25] propose an optimization problem that minimizes the following cost function:

$$J = \left\| \mathbf{f}_i^{des} - \begin{bmatrix} -(C_{\phi^d} S_{\theta^d} C_{\psi^d} + S_{\phi^d} S_{\psi^d}) \\ -(C_{\phi^d} S_{\theta^d} S_{\psi^d} - S_{\phi^d} C_{\psi^d}) \\ -(C_{\phi^d} C_{\theta^d}) \end{bmatrix} T_{total}^{des} \right\|^2 \quad (3.10)$$

With the addition of a weighting matrix, the optimization problem can be summarized as follows:

$$\begin{aligned} \min : J = & \left\| \mathbf{W}_p \left(\mathbf{f}_i^{des} - \begin{bmatrix} -(C_{\phi^d} S_{\theta^d} C_{\psi^d} + S_{\phi^d} S_{\psi^d}) \\ -(C_{\phi^d} S_{\theta^d} S_{\psi^d} - S_{\phi^d} C_{\psi^d}) \\ -(C_{\phi^d} C_{\theta^d}) \end{bmatrix} T_{total}^{des} \right) \right\|^2 \\ \text{s.t. : } & \begin{bmatrix} \psi_{min}^d \\ \theta_{min}^d \\ \phi_{min}^d \\ 4T_{min} \end{bmatrix} \leq \begin{bmatrix} \psi^d \\ \theta^d \\ \phi^d \\ T_{total}^{des} \end{bmatrix} \leq \begin{bmatrix} \psi_{max}^d \\ \theta_{max}^d \\ \phi_{max}^d \\ 4T_{max} \end{bmatrix} \end{aligned} \quad (3.11)$$

The minimum and maximum yaw angle constraints are denoted as ψ_{min}^d and ψ_{max}^d . This notation follows for the pitch and roll angles as well. The minimum and maximum thrusts capable of being generated by a single motor are denoted by T_{min} and T_{max} . The weight-

ing matrix, $\mathbf{W}_p \in \mathbb{R}^{3 \times 3}$, is a diagonal and positive definite matrix, in which the entries define the relative prioritization between the control efforts $f_{i,x}^{des}$, $f_{i,y}^{des}$, and $f_{i,z}^{des}$. The position controller which uses the control allocation approach in (3.11) is termed the ‘optimal position controller’ in this thesis.

The optimization problem in (3.11) can be solved using numerical methods to obtain the desired attitude $(\psi^d, \theta^d, \phi^d)$ and total thrust, T_{total}^{des} , within their defined constraints. Unlike the cost function in the optimal mixer, the cost function in (3.11) is neither quadratic nor convex. Thus, the optimization problem cannot be formulated as a quadratic program (QP) like the optimal mixer. In [25], the Active Set method in sequential quadratic programming (SQP) is used instead. This algorithm requires an initial guess as a starting point for its iterations; the analytical solution obtained from the conventional position controller was used as the initial guess. Similar to the conventional position controller, we use (2.25), with $\mathbf{C}(\psi^d, \theta^d, \phi^d)$ and T_{total}^{des} obtained from (3.11), to compute the desired total thrust that considers the actual attitude of the vehicle, T_{total}^{body} .

An interesting property of the optimal position controller is that, by contrast to the conventional method, the yaw setpoint, ψ^d , need not be specified by the motion planner (as suggested in Fig. 1.2). Instead, ψ^d is determined within the position controller. In order to distinguish between these two cases, the superscript, d , is used to denote that the yaw setpoints come from the position controller, whereas the superscript, ref , indicates that the setpoints come from the motion planner. If it is desired to retain the determination of the yaw setpoints within the motion planner, ψ^{ref} can be specified in the cost function directly and its corresponding constraints removed in (3.11).

It is also worth mentioning that the SQP algorithm used for the optimal position controller in our simulations is much more computationally-expensive and difficult to implement relative to solving a QP problem, as is done for the optimal mixer. Thus, the algorithm may be too slow for real-time applications. A heuristic control allocation method may need to be developed for the optimal position controller to improve computational speeds, similar to the heuristic mixers proposed in [4] and [10] for thrust mixing.

Enabling relative prioritization between the control efforts $f_{i,x}^{des}$, $f_{i,y}^{des}$, and $f_{i,z}^{des}$ allows us to prioritize between the North (x), East (y) position, and altitude control ($-z$), respectively, while respecting attitude and actuator constraints. To demonstrate this capability, we first run the same simulation conducted in Sec. 3.2, but with the conventional position controller replaced with the optimal version subject to an identity weighting matrix. With this matrix, the position controller prioritizes North, East, and altitude control equally. The results of this simulation are plotted in Fig. 3.8, and simulation results in Fig. 3.4 (where the conventional position controller is used) are repeated in this figure for comparison.

In this figure, it can be seen that the vehicle sees a large improvement in altitude control when using the optimal position controller: the maximum deviation from the reference goes from a 70 m to 12 m drop in altitude. The North (x) and East (y) position tracking performance also sees significant improvement with the optimal position controller. A likely explanation behind this improvement can be attributed to the improved attitude control as a consequence of more intelligent handling of attitude constraints by the optimal position controller. Comparing the attitude data in both cases, we can see that, at $t = 13s$, the roll angles begin to saturate for the case with the conventional position controller, but not for the case with the optimal version. This suggests that the optimal position controller is more capable of producing desired attitudes that are within the bounds of the attitude constraints and avoiding saturation, unlike the conventional position controller which can violate the constraints initially, then simply saturates the values at these constraints. Furthermore, towards the end of the maneuver ($t \geq 15s$), the optimal position controller appears to also be capable of producing desired attitudes that are much smoother than those generated by the conventional position controller. This certainly helps in keeping the vehicle's thrust vector more stable, thereby improving tracking performance.

Let us now consider the case where the optimal position controller is forced to prioritize the control effort $f_{i,z}^{des}$ over $f_{i,x}^{des}$ and $f_{i,y}^{des}$. In other words, altitude control is prioritized over North and East position control. To do this, we can run another simulation using

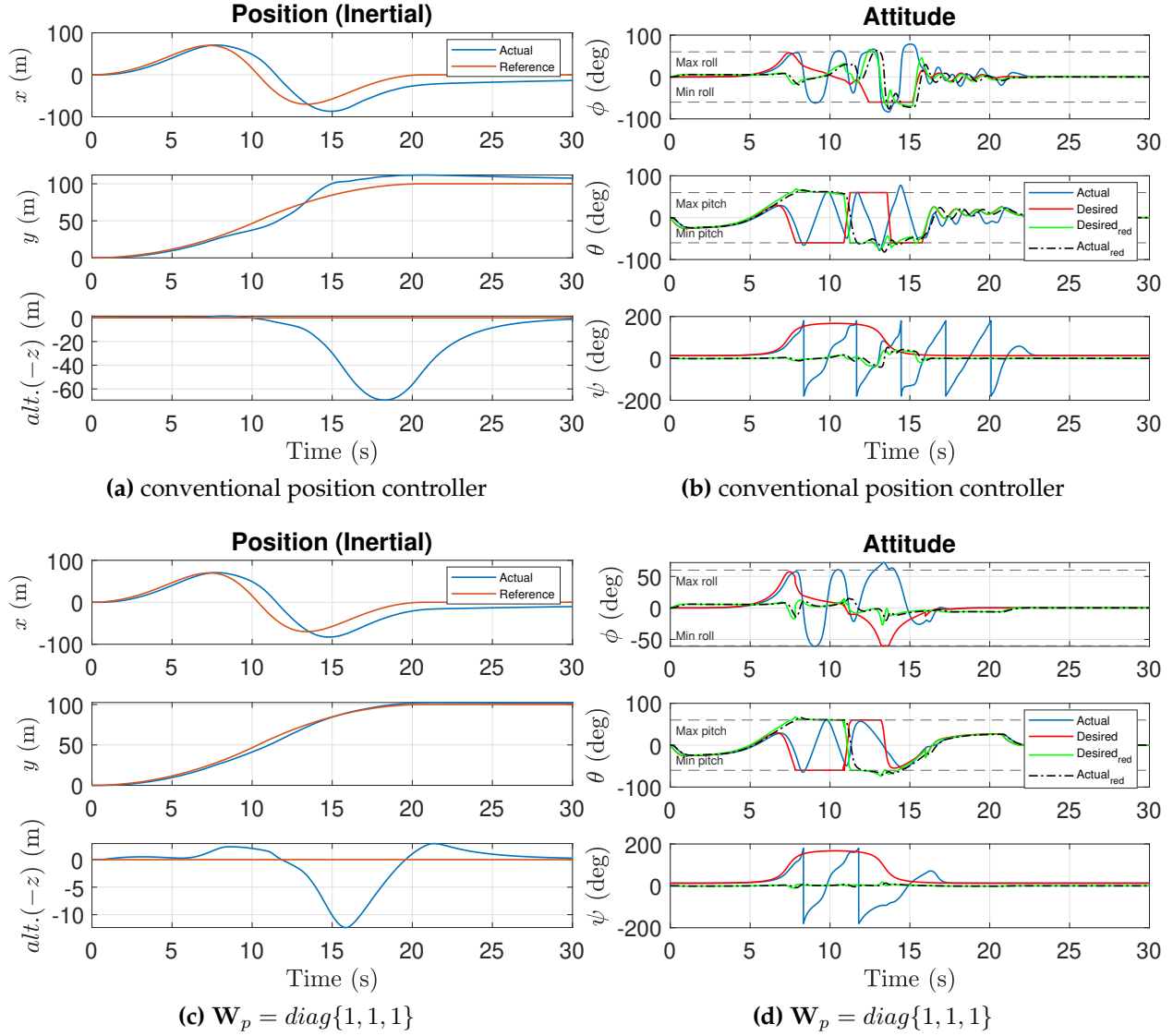


Figure 3.8: Simulation plots with (bottom) and without (top) optimal position controller ($W_p = \text{diag}\{1, 1, 1\}$), decoupled attitude controller, optimal mixer ($W_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

the following weighting matrix: $\mathbf{W}_p = \text{diag}\{1, 1, 100\}$. The weights, in this case, were not chosen from open-loop testing like with the optimal mixer, but were instead obtained by trial and error, since the consistency in their units makes this process more tractable.

The results of this simulation are plotted in Fig. 3.9 (simulation results with $\mathbf{W}_p = \text{diag}\{1, 1, 1\}$ are repeated for comparison). In this figure, we see greater improvement in the altitude control with the weighting matrix that prioritizes it, as expected: the maximum deviation from the reference goes from a 12 m drop in altitude, to about a 4 m increase. Furthermore, the North and East position control performance are worsened due to the new weighting matrix, as expected. This degradation in control is reflected in the attitude data, where we see more erratic control in the roll angle for the new weighting matrix.

Finally, let us now consider the case where the North and East position control are given priority over the altitude control. To do this, we can run another simulation using the following weighting matrix: $\mathbf{W}_p = \text{diag}\{100, 100, 1\}$. Simulation results with this matrix are plotted in Fig. 3.10 (simulation results with $\mathbf{W}_p = \text{diag}\{1, 1, 1\}$ are repeated for comparison). In this figure, we see that the altitude control does diminish with the altitude dropping to around 25 m from 12 m, which is expected given that altitude control is the least prioritized in the weighting matrix. As for the North and East position control, improvements in tracking appear to be minimal relative to the degradation in altitude control. Interestingly, when comparing this simulation to the one conducted in Sec. 3.2 (with results plotted in Fig. 3.4) where the conventional position controller is used, the altitude control appears to have performed better with the optimal position controller despite the weighting matrix used: with the optimal position controller, the altitude drops to at most 25 m, whereas it drops to 70 m with the conventional position controller. This, too, can be explained by the improved attitude control as a consequence of more intelligent handling of the actuator constraints by the optimal position controller. Comparing the attitude plots of both cases, we see, again, much lower duration in the saturation times of the roll and pitch (but mostly roll) angles with the optimal position controller. Moreover, the desired attitudes outputted by the optimal position controller appear to be

far smoother and less erratic. Overall, these benefits improve the stability and control of the thrust vector, leading to improvements in position control in all three directions.

With the optimal position controller, in combination with a decoupled attitude controller and optimal mixer, we were able to substantially improve the stability and control of the vehicle relative to the conventional approach. Notably, we were able to significantly improve the altitude control in order to reduce the risk of a potential collision with the ground. The proposed methods offer greater flexibility in choosing which vehicle states should be prioritized for tracking in the event of saturation. Collectively, these methods are a unified approach to addressing actuator and attitude constraints at every level of the control system, excluding the motion planner.

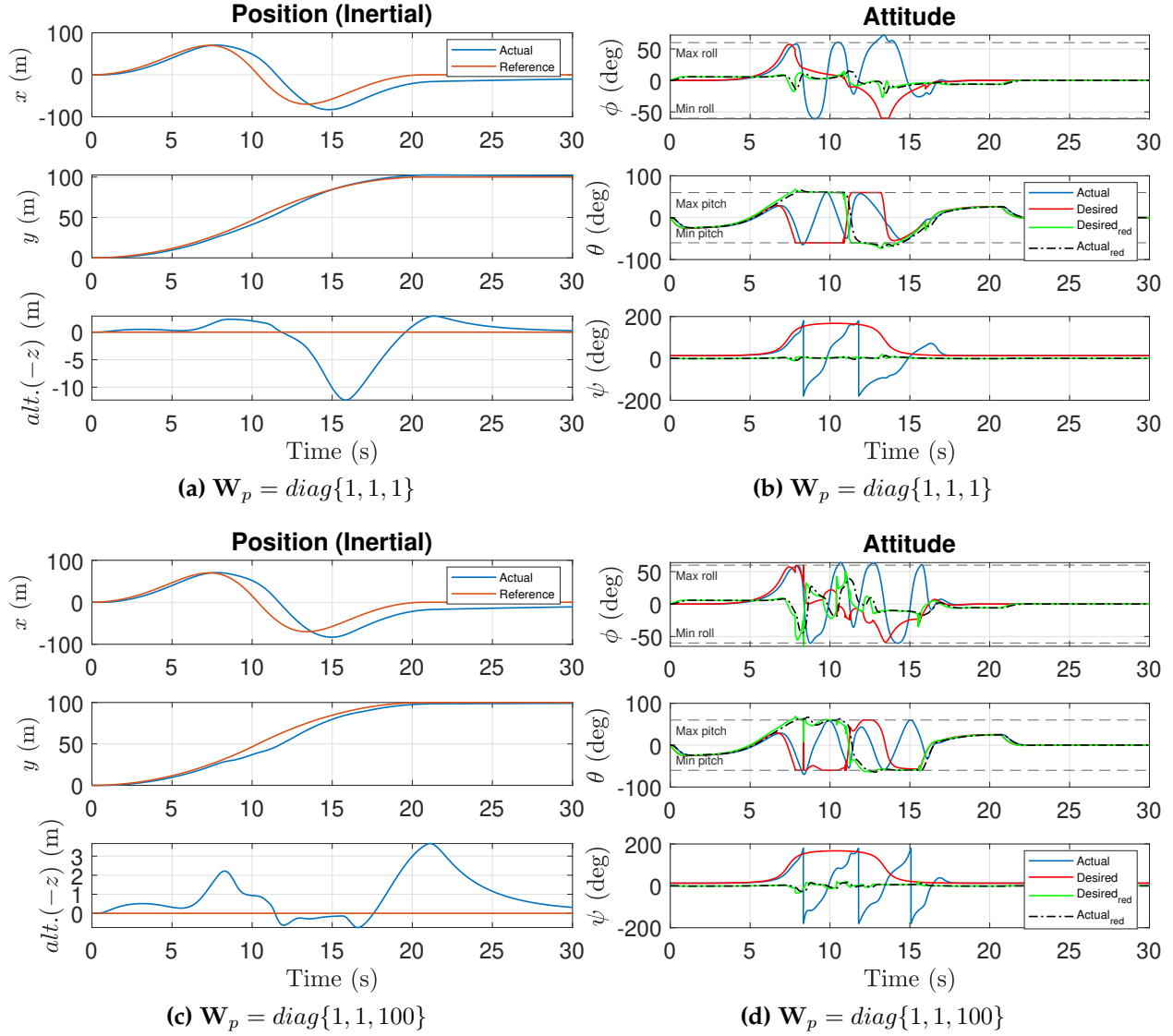


Figure 3.9: Simulation plots for optimal position controller (with different W_p 's), decoupled attitude controller, optimal mixer ($W_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

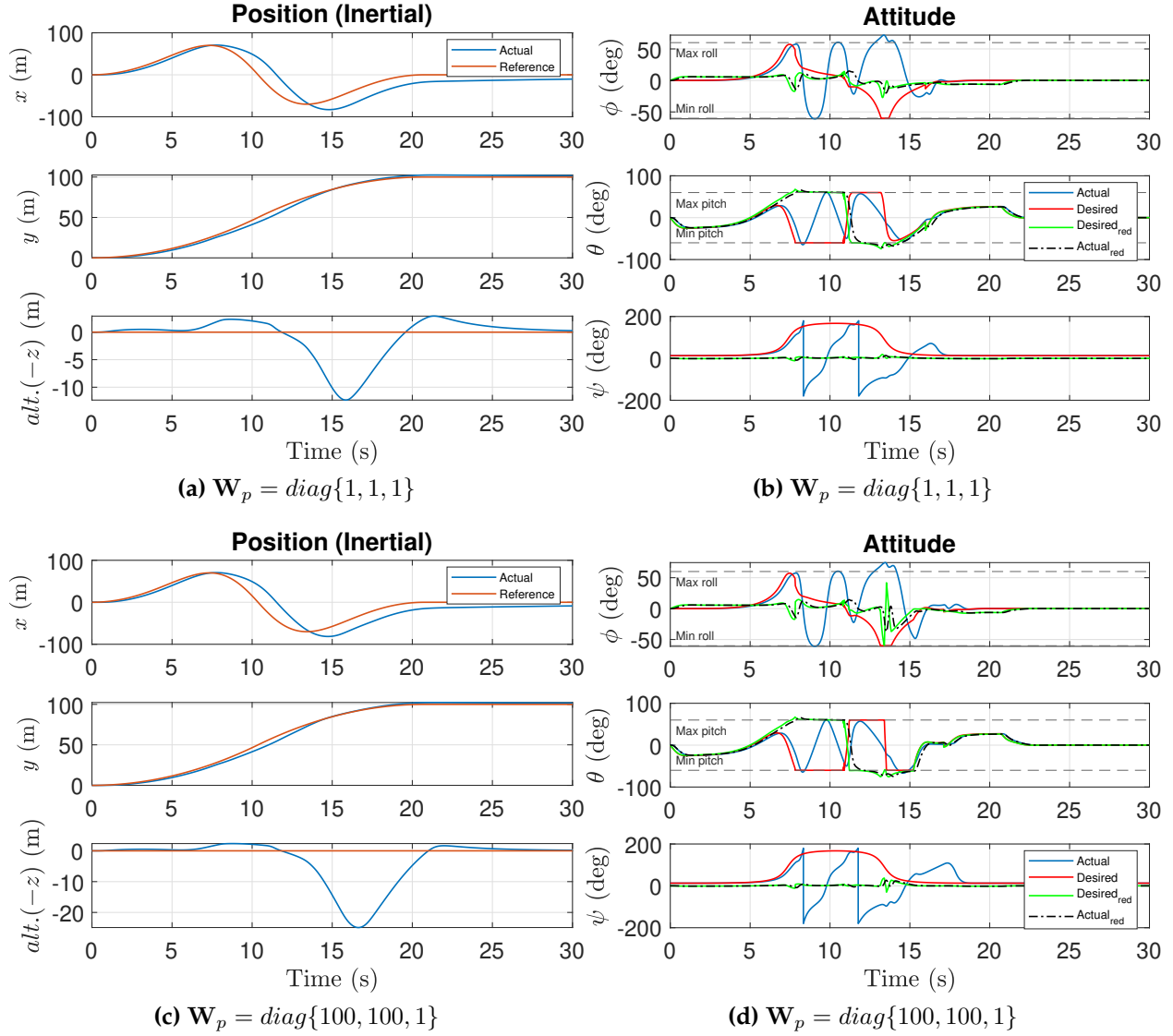


Figure 3.10: Simulation plots for optimal position controller (with different W_p 's), decoupled attitude controller, optimal mixer ($W_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

Chapter 4

Motion Planning

This chapter will explore the role of motion planning in generating trajectories that avoid actuator saturation. This is a distinct role from the previous subsystems which were responsible for ‘handling’ the saturation when it occurs, rather than avoiding it. In addition to generating trajectories that are within actuator constraints, the motion planner should also satisfy any imposed state constraints, such as constraints on maximum velocity, acceleration, attitude etc.

4.1 Maneuver Description

This thesis investigates the use of a waypoints-based motion planner responsible for generating trajectories that avoid actuator saturation for the purposes of performing area coverage. Area coverage refers to traversing and scanning an area with a sensor (such as an onboard camera) attached to the quadrotor [42]. This type of drone operation was chosen as it is one of the most common operations performed by quadrotors [42–44]. More specifically, this thesis investigates motion planning for area coverage by generating trajectories that mimic the ‘lawnmower maneuver’, which is considered to be the most conventional maneuver for area coverage [42, 43]. The popularity of this maneuver stems

from its simplicity and effectiveness at traversing long distances in open areas (without obstacles) while acquiring sensor data uniformly [45]. The uniformity of the collected data makes ‘stitching’ the data together much more convenient.

A segment of the lawnmower maneuver to be examined is shown in Fig. 4.1. In this figure, there are four waypoints, indicated by blue crosses, where waypoint 1 is at the start of the maneuver and waypoint 4 is its end. This segment can be repeated, infinitely many times, to survey an arbitrary rectangular area with a desired level of spacing between survey lines. These waypoints are placed at arbitrary distances of 10 m , 100 m , and 0 m in the Northern (x), Eastern (y), and Down (z) directions, respectively, in the inertial frame. The red line is the position trajectory, $\mathbf{p}_i^{ref}(t)$ (expressed in the inertial frame, as denoted by subscript i), which passes through all of these waypoints and is divided into three segments: $\mathbf{p}_{i,1}^{ref}(t)$, $\mathbf{p}_{i,2}^{ref}(t)$, $\mathbf{p}_{i,3}^{ref}(t)$. The first and third segments must be straight lines in accordance with the lawnmower maneuver, with the second segment corresponding to a u-turn. The turn radius and general shape of this u-turn are variables that the motion planner will need to optimize. In this thesis, we will optimize this u-turn for the shortest time needed to complete the whole maneuver.

Each of the three segments consists of three 9th order polynomial trajectories for the inertial North, East, and Down positions with unknown coefficients. An example is shown here for the first segment:

$$\mathbf{p}_{i,1}^{ref}(t) = \begin{bmatrix} p_{i,x_1}^{ref}(t) \\ p_{i,y_1}^{ref}(t) \\ p_{i,z_1}^{ref}(t) \end{bmatrix} = \begin{bmatrix} c_0 + c_1t + c_2t^2 + \dots + c_9t^9 \\ c_{10} + c_{11}t + c_{12}t^2 + \dots + c_{19}t^9 \\ c_{20} + c_{21}t + c_{22}t^2 + \dots + c_{29}t^9 \end{bmatrix} \quad (4.1)$$

The coefficients for all three segments can be collected accordingly:

$$\mathbf{c}_1 = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{29} \end{bmatrix}, \mathbf{c}_2 = \begin{bmatrix} c_{30} \\ c_{31} \\ \vdots \\ c_{59} \end{bmatrix}, \mathbf{c}_3 = \begin{bmatrix} c_{60} \\ c_{61} \\ \vdots \\ c_{89} \end{bmatrix} \quad (4.2)$$

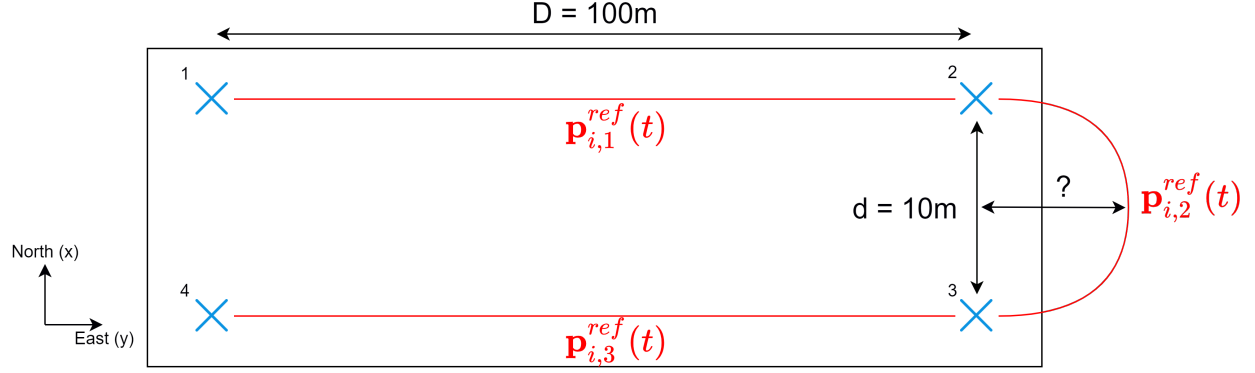


Figure 4.1: Top view of Lawnmower maneuver.

The reasoning behind these polynomials being 9^{th} order will be discussed in the next section. The coefficient values will determine the spatial and temporal properties of their respective segments. In order to solve for these coefficients analytically, the boundary conditions of each segment must be fully defined up to the 4^{th} derivative of position. This analytical approach is not practically reasonable because many of the boundary conditions are usually unknown to the motion planner. For example, while the position and velocity may be known at waypoint 1, the acceleration and jerk (3^{rd} derivative of position) are less likely to be known. Moreover, even less information is likely to be known at waypoints 2 and 3, beyond their positions. Thus, a new approach is needed.

4.2 Minimum Snap Trajectory Generation

Minimum snap trajectory generation is an optimal approach to finding the polynomial coefficients such that the snap of the vehicle (4^{th} derivative of position) is minimized. The snap is chosen for minimization instead of, say, the acceleration because with a cascaded control structure, the position control system is a 4^{th} order system [24]. Therefore, in order to produce trajectories with minimum control effort, the snap, not the acceleration, must

be minimized. The cost function for minimizing the snap can be formulated as:

$$J = \int_0^{T_1} \left\| \frac{d^4 \mathbf{P}_{i,1}^{ref}}{dt^4} \right\|^2 dt + \int_0^{T_2} \left\| \frac{d^4 \mathbf{P}_{i,2}^{ref}}{dt^4} \right\|^2 dt + \int_0^{T_3} \left\| \frac{d^4 \mathbf{P}_{i,3}^{ref}}{dt^4} \right\|^2 dt \quad (4.3)$$

where T_i is the time duration of the i^{th} segment, and $\|\cdot\|$ is the Euclidean norm of the vector in (\cdot) . The time durations, T_i , must be specified in order to solve the cost minimization problem. These times may be known to the vehicle operator depending on the vehicle's mission, but can otherwise be estimated using an average vehicle velocity traversing through the segments of known distance. The integrals in the cost function can now be expanded and further simplified, as follows:

$$J = \mathbf{c}_1^T \mathbf{Q}_1 \mathbf{c}_1 + \mathbf{c}_2^T \mathbf{Q}_2 \mathbf{c}_2 + \mathbf{c}_3^T \mathbf{Q}_3 \mathbf{c}_3 \quad (4.4)$$

where \mathbf{Q}_i is found using the following equations from [26]:

$$\mathbf{M}^{r,c}(T) = \begin{cases} 2 \frac{T^{r+c-2n+1}}{r+c-2n+1} \prod_{m=0}^{n-1} (r-m)(c-m) & \text{for } r \geq n \wedge c \geq n \\ 0 & \text{for } r < n \vee c < n \end{cases} \quad (4.5)$$

$$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{M}(T_1) & 0 \dots 0 \\ 0 \dots \mathbf{M}(T_1) & 0 \dots 0 \\ 0 \dots 0 \dots \mathbf{M}(T_1) \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} \mathbf{M}(T_2) & 0 \dots 0 \\ 0 \dots \mathbf{M}(T_2) & 0 \dots 0 \\ 0 \dots 0 \dots \mathbf{M}(T_2) \end{bmatrix}, \quad \mathbf{Q}_3 = \begin{bmatrix} \mathbf{M}(T_3) & 0 \dots 0 \\ 0 \dots \mathbf{M}(T_3) & 0 \dots 0 \\ 0 \dots 0 \dots \mathbf{M}(T_3) \end{bmatrix} \quad (4.6)$$

where $\mathbf{M}^{r,c}$ refers to the element of $\mathbf{M} \in \mathbb{R}^{10 \times 10}$ in row r and column c . The value, n , refers to the number of states we are generating trajectories for. In this case, $n = 3$ since we are considering trajectories for three states/directions: x , y , and z . The first term in (4.4) corresponds to the first term in (4.3), and this follows for the other terms accordingly. The simplification of the cost function is described in more detail in [26], but to summarize some key points: The dimension of the square matrix, \mathbf{M} , depends on the order of the polynomials being used (9^{th} order in this case), plus one. The number of \mathbf{M} 's found in the diagonal of \mathbf{Q}_i is equal to n .

According to [26], the cost function in (4.4) can be simplified further, as follows:

$$J = \mathbf{c}_1^T \mathbf{Q}_1 \mathbf{c}_1 + \mathbf{c}_2^T \mathbf{Q}_2 \mathbf{c}_2 + \mathbf{c}_3^T \mathbf{Q}_3 \mathbf{c}_3 = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_1 & 0 \dots & 0 \dots \\ 0 \dots & \mathbf{Q}_2 & 0 \dots \\ 0 \dots & 0 \dots & \mathbf{Q}_3 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix} = \underline{\mathbf{c}}^T \underline{\mathbf{Q}} \underline{\mathbf{c}} \quad (4.7)$$

$$\underline{\mathbf{c}} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}, \quad \underline{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_1 & 0 \dots & 0 \dots \\ 0 \dots & \mathbf{Q}_2 & 0 \dots \\ 0 \dots & 0 \dots & \mathbf{Q}_3 \end{bmatrix} \quad (4.8)$$

Constraints must be specified for this optimization problem including boundary conditions, which are known beforehand, as well as continuity constraints. Continuity constraints ensure continuity of 0^{th} through 4^{th} derivatives of position at the endpoints of each segment. Ninth order polynomials are used for the trajectories because they have 10 coefficients, the minimum number needed for the continuity constraints. All of these constraints can be formulated as equality constraints, and so the complete optimization problem is as follows:

$$\begin{aligned} \min : J &= \underline{\mathbf{c}}^T \underline{\mathbf{Q}} \underline{\mathbf{c}} \\ \text{s.t.} : \mathbf{A} \underline{\mathbf{c}} &= \mathbf{b} \end{aligned} \quad (4.9)$$

Given the equality constraints, this optimization problem has an analytical solution, as discussed in detail in [26]. However, if the boundary conditions do not need to be specified exactly (e.g., the velocity at waypoint 4 is greater than 0 m/s, not equal to 0 m/s), then inequality constraints can be used instead [24]. In this more general case, the optimization problem becomes a quadratic program (or QP) which can be solved efficiently using, for example, an Active Set quadratic programming algorithm like in [15].

Note that the motion planner is also responsible for supplying a yaw trajectory, $\psi^{ref}(t)$, as indicated in Fig. 1.2. Like the position trajectory, the yaw trajectory can be divided into

three segments and (4.1) can be modified as such:

$$\mathbf{o}_{i,1}^{ref}(t) = \begin{bmatrix} p_{i,x_1}^{ref}(t) \\ p_{i,y_1}^{ref}(t) \\ p_{i,z_1}^{ref}(t) \\ \psi_1^{ref}(t) \end{bmatrix} = \begin{bmatrix} c_0 + c_1t + c_2t^2 + \dots + c_9t^9 \\ c_{10} + c_{11}t + c_{12}t^2 + \dots + c_{19}t^9 \\ c_{20} + c_{21}t + c_{22}t^2 + \dots + c_{29}t^9 \\ c_{30} + c_{31}t + c_{32}t^2 + \dots + c_{39}t^9 \end{bmatrix} \quad (4.10)$$

The new cost function, according to [26], is as follows:

$$J = \int_0^{T_1} \left\| \frac{d^4 \mathbf{o}_{i,1}^{ref}}{dt^4} \right\|^2 dt + \int_0^{T_2} \left\| \frac{d^4 \mathbf{o}_{i,2}^{ref}}{dt^4} \right\|^2 dt + \int_0^{T_3} \left\| \frac{d^4 \mathbf{o}_{i,3}^{ref}}{dt^4} \right\|^2 dt \quad (4.11)$$

With this new cost function, the problem maintains its QP nature and can be solved equivalently using $n = 4$ to find \mathbf{Q}_i . Note that we are now including an angle alongside positions in this new cost function. This mismatch of units may require using gains to nondimensionalize the cost function, as is done in [24]. In [26], however, the units were assumed to be equivalent and this assumption was found to produce adequate trajectories. For this thesis, we will also make this assumption due to the compactness and simplicity of its resultant cost function.

4.3 Differential Flatness

With the position and yaw trajectories obtained from the QP problem, we need a method of determining whether the trajectories are within our defined actuator and state constraints. We know that the quadrotor's dynamics, defined in Chapter 2 of this thesis, are differentially flat. Differential flatness provides an analytical mapping between selected flat outputs and their derivatives to the states and control input required to track that trajectory [24]. We can use the differential flatness properties to determine all of our vehicle's states and inputs for the computed trajectories, and hence determine if there are any constraint violations. In the event of violations, the trajectories can subsequently be

modified to mitigate them. Our choice of flat outputs are the vehicle's inertial position and yaw ($p_{i,x}$, $p_{i,y}$, $p_{i,z}$, and ψ). In the following subsections, we will describe the algebraic equations for the vehicle's states and inputs as functions of these flat outputs and their derivatives, collectively referred to as the 'differential flatness equations'. In these equations, the flat outputs will be set to the position and yaw trajectories obtained from the QP problem ($p_{i,x}^{ref}$, $p_{i,y}^{ref}$, $p_{i,z}^{ref}$, and ψ^{ref}) as these are the desired states of our vehicle.

4.3.1 Position, Velocity, and Acceleration

As stated previously, the coefficients of the position trajectory segments $\mathbf{p}_{i,1}^{ref}$, $\mathbf{p}_{i,2}^{ref}$, and $\mathbf{p}_{i,3}^{ref}$, are obtained from the QP problem. To find the inertial velocity and acceleration of the vehicle, we can take the 1st and 2nd derivative of each segment, respectively. These position segments are collectively the overall position trajectory, \mathbf{p}_i^{ref} , by definition. For convenience, we will formulate the differential flatness equations treating \mathbf{p}_i^{ref} as a single continuous segment to avoid confusion using multiple segments.

4.3.2 Thrust and Attitude

The coefficients of the yaw trajectory segments, ψ_1^{ref} , ψ_2^{ref} , and ψ_3^{ref} , are also obtained from the QP problem. Similarly, we will treat ψ^{ref} as a single continuous segment for convenience. We can use this information alongside the inertial acceleration to find the total thrust needed throughout the maneuver, as well as the pitch and roll angles using (2.22)–(2.24):

$$T_{total} = m \sqrt{(\ddot{p}_{i,x}^{ref})^2 + (\ddot{p}_{i,y}^{ref})^2 + (\ddot{p}_{i,z}^{ref} - g)^2} \quad (4.12)$$

$$\theta^d = \arctan \frac{\cos \psi^{ref} \ddot{p}_{i,x}^{ref} + \sin \psi^{ref} \ddot{p}_{i,y}^{ref}}{\ddot{p}_{i,z}^{ref} - g} \quad (4.13)$$

$$\phi^d = \arcsin \frac{\cos \psi^{ref} \ddot{p}_{i,y}^{ref} - \sin \psi^{ref} \ddot{p}_{i,x}^{ref}}{\sqrt{(\ddot{p}_{i,x}^{ref})^2 + (\ddot{p}_{i,y}^{ref})^2 + (\ddot{p}_{i,z}^{ref} - g)^2}} \quad (4.14)$$

Note that g , the acceleration due to gravity, is subtracted from $\ddot{p}_{i,z}^{ref}$. This is to ensure that there is a thrust component which counters the weight of the vehicle and its effect on the acceleration is cancelled. In other words, the vehicle would only track $\ddot{p}_{i,z}^{ref}$.

4.3.3 Angular Velocity

The following equations are taken directly from [31] which discusses differential flatness of quadrotor dynamics with rotor drag, but without the rotor drag terms. The frames used in [31] differ slightly from the ones used in this thesis. More specifically, the authors use the East-North-Up (ENU) convention for their inertial frame, defined by the basis vectors $\{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$. The body frame is given by the vectors $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$, and an intermediate frame is given by the vectors $\{\mathbf{x}_C, \mathbf{y}_C, \mathbf{z}_C\}$. This intermediate frame is simply the inertial frame rotated about \mathbf{z}_W by a yaw angle ψ . The following equations have been modified to accommodate the frames used in this thesis, as described in Fig. 2.1, and the same notations from [31] are used here for consistency.

We can compute the basis vectors defined in the inertial frame as functions of the position and yaw trajectories as follows:

$$\mathbf{x}_C = \begin{bmatrix} \cos \psi^{ref} & \sin \psi^{ref} & 0 \end{bmatrix}^T \quad (4.15)$$

$$\mathbf{y}_C = \begin{bmatrix} -\sin \psi^{ref} & \cos \psi^{ref} & 0 \end{bmatrix}^T \quad (4.16)$$

$$\mathbf{z}_C = \mathbf{z}_W = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (4.17)$$

$$\boldsymbol{\alpha} = \ddot{\mathbf{p}}_i^{ref} - g\mathbf{z}_W \quad (4.18)$$

$$\mathbf{x}_B = \frac{\mathbf{y}_C \times \boldsymbol{\alpha}}{\|\mathbf{y}_C \times \boldsymbol{\alpha}\|} \quad (4.19)$$

$$\mathbf{y}_B = \frac{\boldsymbol{\alpha} \times \mathbf{x}_B}{\|\boldsymbol{\alpha} \times \mathbf{x}_B\|} \quad (4.20)$$

$$\mathbf{z}_B = \mathbf{x}_B \times \mathbf{y}_B \quad (4.21)$$

With these basis vectors, we can compute the angular velocities:

$$c = \frac{T_{total}}{m} = \mathbf{z}_B^T (\ddot{\mathbf{p}}_i^{ref} - g\mathbf{z}_W) \quad (4.22)$$

$$\mathbf{j} = \frac{d^3 \mathbf{p}_i^{ref}}{dt^3} \quad (4.23)$$

$$\boldsymbol{\omega}_b = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \quad (4.24)$$

$$\omega_x = -\frac{\mathbf{y}_B^T \mathbf{j}}{c} \quad (4.25)$$

$$\omega_y = \frac{\mathbf{x}_B^T \mathbf{j}}{c} \quad (4.26)$$

$$\omega_z = \frac{\dot{\psi}^{ref} \mathbf{x}_C^T \mathbf{x}_B + \omega_y \mathbf{y}_C^T \mathbf{z}_B}{\|\mathbf{y}_C \times \mathbf{z}_B\|} \quad (4.27)$$

where \mathbf{j} is the jerk vector and $\boldsymbol{\omega}_b$ is the vehicle's angular velocity vector resolved in the body frame.

4.3.4 Angular Acceleration

The angular accelerations along each body axis can also be found using the following equations from [31]:

$$\dot{c} = \mathbf{z}_B^T \mathbf{j} \quad (4.28)$$

$$\mathbf{s} = \frac{d^4 \mathbf{p}_i^{ref}}{dt^4} \quad (4.29)$$

$$\dot{\boldsymbol{\omega}}_b = \begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y & \dot{\omega}_z \end{bmatrix}^T \quad (4.30)$$

$$\dot{\omega}_x = -\frac{\mathbf{y}_B^T \mathbf{s} - 2\dot{c}\omega_x + c\omega_y\omega_z}{c} \quad (4.31)$$

$$\dot{\omega}_y = \frac{\mathbf{x}_B^T \mathbf{s} - 2\dot{c}\omega_y - c\omega_x\omega_z}{c} \quad (4.32)$$

$$\dot{\omega}_z = \frac{\ddot{\psi}^{ref} \mathbf{x}_C^T \mathbf{x}_B + 2\dot{\psi}^{ref} \omega_z \mathbf{x}_C^T \mathbf{y}_B - 2\dot{\psi}^{ref} \omega_y \mathbf{x}_C^T \mathbf{z}_B - \omega_x \omega_y \mathbf{y}_C^T \mathbf{y}_B - \omega_x \omega_z \mathbf{y}_C^T \mathbf{z}_B + \dot{\omega}_y \mathbf{y}_C^T \mathbf{z}_B}{\|\mathbf{y}_C \times \mathbf{z}_B\|} \quad (4.33)$$

where \mathbf{s} is the snap vector and $\dot{\boldsymbol{\omega}}_b$ is the vehicle's angular acceleration vector resolved in the body frame.

4.3.5 Moments and Motor Thrusts

Using the rigid body model for our quadrotor, the moments imparted on the vehicle can be found as functions of the angular velocity and acceleration vectors:

$$\mathbf{m}_b = \mathbf{J}_b \dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \mathbf{J}_b \boldsymbol{\omega}_b \quad (4.34)$$

where \mathbf{J}_b and \mathbf{m}_b refer to the vehicle's second moment of mass and net external moment vector, respectively. Finally, the individual motor thrusts can be calculated using the conventional mixer (see Sec. 2.2.3).

With the position and yaw trajectories obtained from the QP problem, we can now use the differential flatness equations from (4.12) to (4.34) to find, in closed form, all relevant variables of the quadrotor flight. Variables include states and inputs, such as linear and angular velocity, linear and angular acceleration, attitude, and individual motor thrusts. Thus, in cases where we have constraints on certain states and inputs, we can use these equations to identify any constraint violations that may occur during flight.

4.4 Time Scaling

Time scaling, or temporal scaling, is the process by which the time history of a trajectory can be scaled to be more or less aggressive while maintaining its spatial properties [24]. This is a particularly useful method for allowing us to modify our trajectories if constraint violations are found via the differential flatness equations. To best illustrate this method, let's take the x -component of the first segment acquired from the QP problem, $p_{i,x_1}^{ref}(t)$, and

scale its time history by some value, s :

$$\tau = \frac{t}{s} \quad (4.35)$$

$$p_{i,x_1}^{ref,scaled}(\tau) = c_0 + c_1\tau + c_2\tau^2 + \dots + c_9\tau^9 \quad (4.36)$$

$$p_{i,x_1}^{ref,scaled}(t) = c_0 + \frac{c_1t}{s} + \frac{c_2t^2}{s^2} + \dots + \frac{c_9t^9}{s^9} \quad (4.37)$$

If s is less than 1, then it takes a shorter amount of time execute the first segment of the trajectory. In other words, the trajectory is now more aggressive. Alternatively, if s is greater than 1, it becomes less aggressive. We can use this scaling method across all segments to change the time history such that our trajectories will satisfy state and actuator constraints while using the differential flatness equations for validation. Finding the appropriate scale that satisfies this condition analytically can prove to be difficult given the inherent nonlinearities in some of the differential flatness equations, like in (4.34). Hence, an iterative approach is required.

Whether the iteration should be done via automation or manually is a design choice left to the flight operator; if trajectory generation is done online, automation is required where s would be iterated over some defined increment until the modified trajectories no longer violate any constraints. On the other hand, if trajectory generation can be done offline, a manual approach can give the flight operator more control over the process, allowing them to consider other vehicle constraints such as flight time (which depends on battery charge).

4.5 Trajectory Generation and Scaling Example

Trajectory generation begins with solving the minimum snap optimization problem. The boundary conditions for this problem were defined as follows: the x and y positions of all waypoints were defined as shown in Fig. 4.1 (with altitude set to 0 m throughout the maneuver), and the velocities at waypoints 1 and 4 were set to 0 m/s so that the vehicle

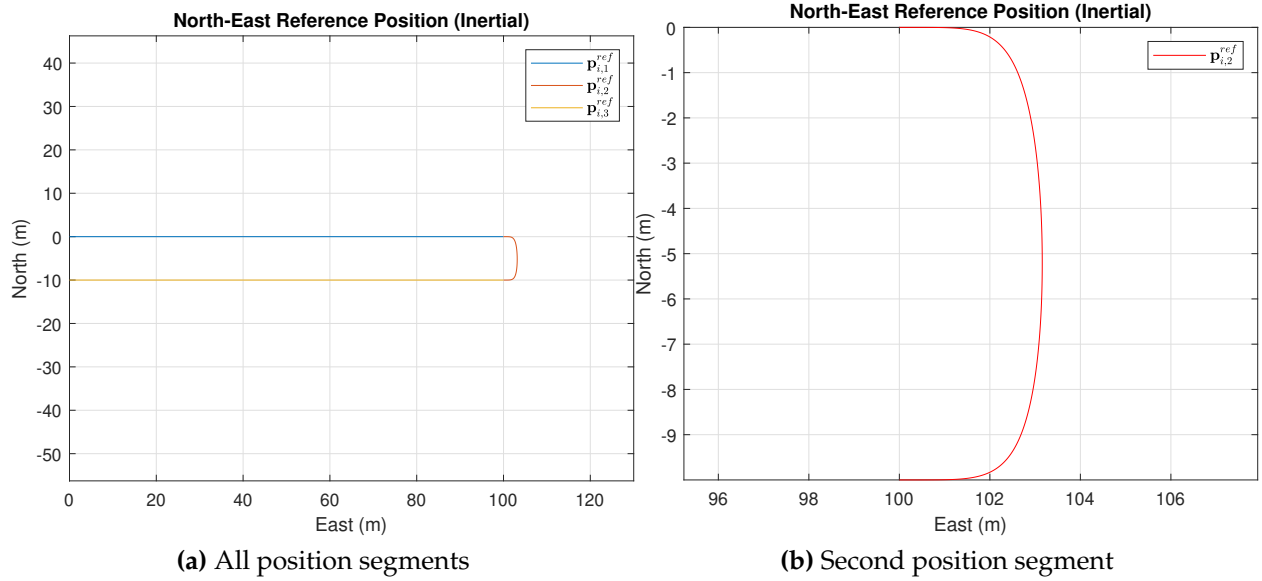


Figure 4.2: Position trajectory given $T_1 = T_3 = 10s$ and $T_2 = 2s$.

starts and ends the maneuver at rest. The segment times T_1 , T_2 , and T_3 also needed to be defined for our cost function, so we followed the advice shared in Sec. 4.2 and assumed some average velocity to estimate the segment times. We assumed that the vehicle can achieve an average velocity of 10 m/s . Given the distances defined in Fig. 4.1, this means that it will take 10 seconds to execute the first segment ($T_1 = 10s$) and this follows for the third segment as well ($T_3 = 10s$) since they are of equal distance. For the second segment, estimating its corresponding segment time, T_2 , is not as simple because it is not a straight line. Therefore, for now, we will set it as 2 seconds ($T_2 = 2s$) and we will discuss in Sec. 4.6 how altering T_2 can change the spatial properties of the second segment. In this example, the yaw was set to zero throughout the maneuver for simplicity (all of its coefficients are zero).

Using these segment times, we were able to solve the QP problem and obtain the position trajectory shown in Fig. 4.2. From this figure, we can see that the maximum Eastern displacement of the second segment, henceforth referred to as the ‘maximum turn displacement’ for convenience, is around 3 m . With the same quadrotor inertia and motor parameters listed in Table 2.2, we can now use the differential flatness equations to plot our vehicle’s states and inputs throughout this trajectory, as shown in Fig. 4.3.

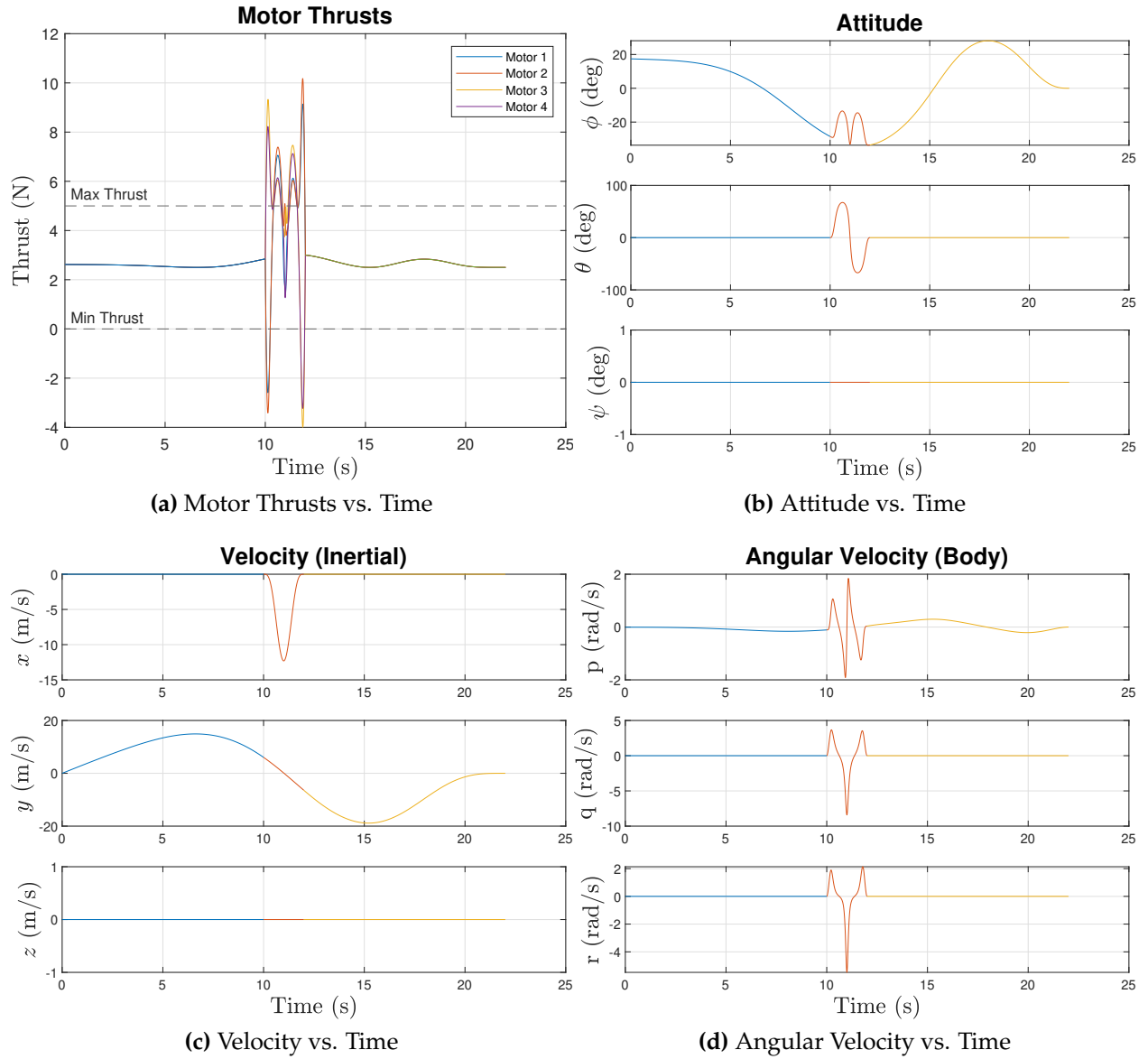


Figure 4.3: Plots obtained from differential flatness equations for the computed trajectory with 4 waypoints.

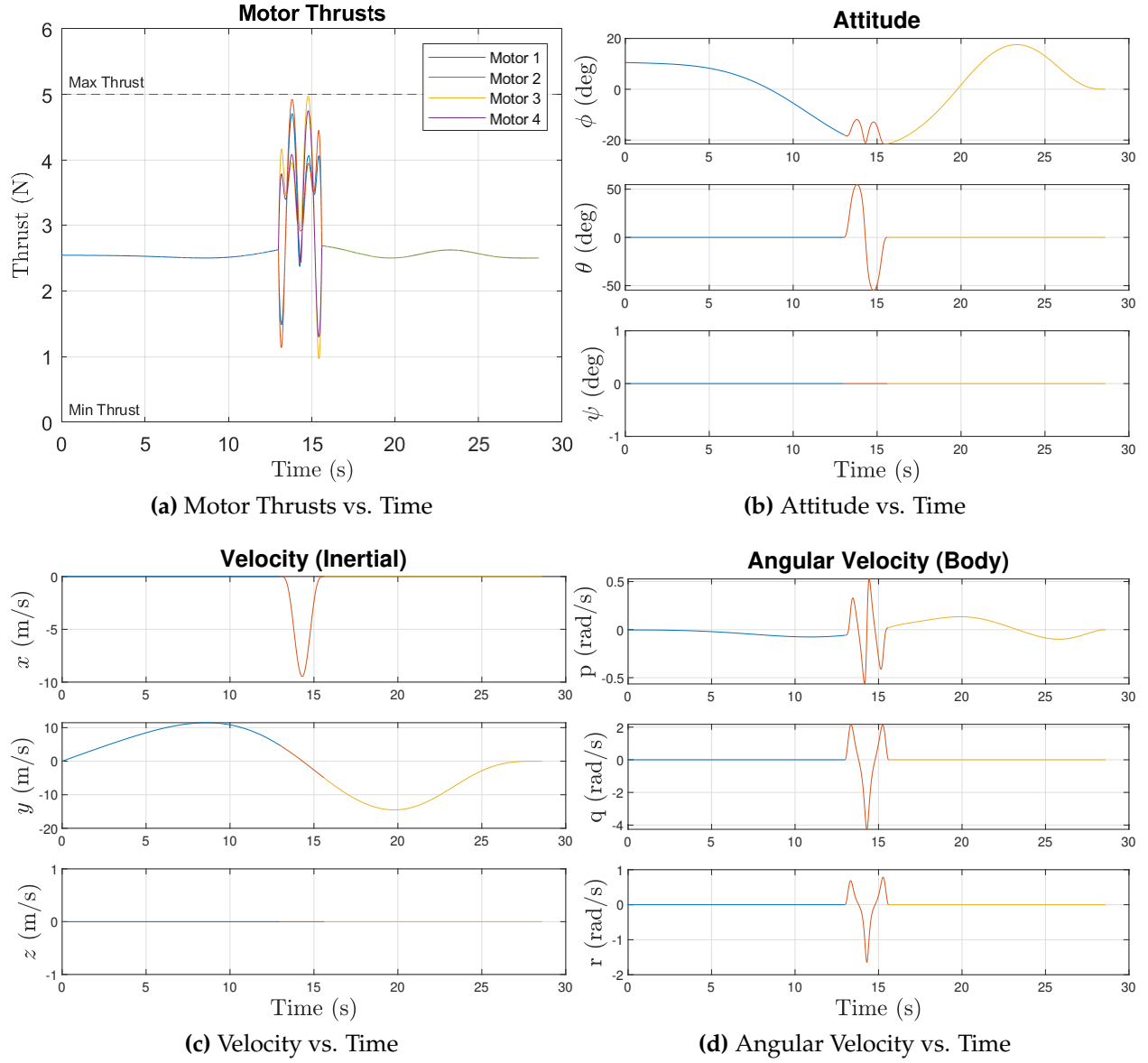


Figure 4.4: Plots obtained from differential flatness equations for the scaled trajectory with 4 waypoints, $s = 1.3$.

A maximum motor thrust constraint of 5 N will be considered in this example and it is shown to be violated by our computed trajectory in this figure. Therefore, iterative time scaling is needed to modify our trajectory and correct this issue.

Manual tuning of the scale, s , was conducted wherein s was increased incrementally to make the maneuver less aggressive until the motor thrusts were within constraints. A scale of 1.3 was found to satisfy this condition, as shown in Fig. 4.4. From this figure, it's clear that all states experience a drop in magnitude by making the maneuver less aggressive, as expected. It should be noted that the trajectory shown in Fig. 4.2 is unaffected by time scaling and this applies also to the scaled maneuver.

4.6 Time Optimal Trajectory Generation

In the previous section, we chose arbitrary segment times and scaled them accordingly to satisfy the actuator constraints. Through scaling, the total time to execute the trajectory went from 22 seconds (as prescribed by the specified durations T_1, T_2, T_3) to 28.6 seconds ($22 * s, s = 1.3$). It was noted earlier in this chapter that we would prefer to have a trajectory that is optimized to minimize the total time, and so we can formulate the following question: How can we select our initial segment times such that the total time is minimized *after* we apply time scaling?

To answer this question, we must first recognize that only the second segment is capable of being spatially modified to minimize the total time if we assume that D and d (shown in Fig. 4.1) are prescribed. In fact, the spatial properties of the second segment is found to depend only on the ratio:

$$\mu = \frac{T_2}{T_1} \tag{4.38}$$

where μ is termed the 'turn ratio'. This statement rests on the assumption that $T_1 = T_3$, which is reasonable given that the first and third segments are similar. Fig. 4.5 illus-

Table 4.1: Turn ratios and total time for maneuver with 4 waypoints

μ	Total time (s)
0.1	50.4
0.2	28.6
0.3	21.85
0.4	19.2
0.5	17.5
0.6	16.9
0.7	16.47
0.8	16.52
0.9	16.53
1	16.8

trates how varying the turn ratio dramatically changes the shape of the second segment; a higher ratio leads to a greater turn displacement.

In order to find the lowest possible total time to execute the whole maneuver, we can look at the total time for a variety of turn ratios, after performing time scaling to use as much thrust from the motors without violating their constraints. The results of this process are listed in Table 4.1. From this table, it can be concluded that a turn ratio of 0.7 yields the minimum total time. The shape of the second segment with this ratio is shown in Fig. 4.5 and the states and inputs associated with the overall trajectory are shown in Fig. 4.6. It is noteworthy that, when comparing the case with $\mu = 0.7$ to the prior one with $\mu = 0.2$, we find that $T_1 = 6.1s$ and $T_2 = 4.27s$ versus $T_1 = 13s$ and $T_2 = 2.6s$, respectively. That is, the average velocity throughout the maneuver is much higher with $\mu = 0.7$, but the maximum thrust needed remains unchanged (of course, this neglects air drag). Comparing Fig. 4.4 and 4.6 shows that the case with $\mu = 0.7$ is generally smoother and less abrupt in transition.

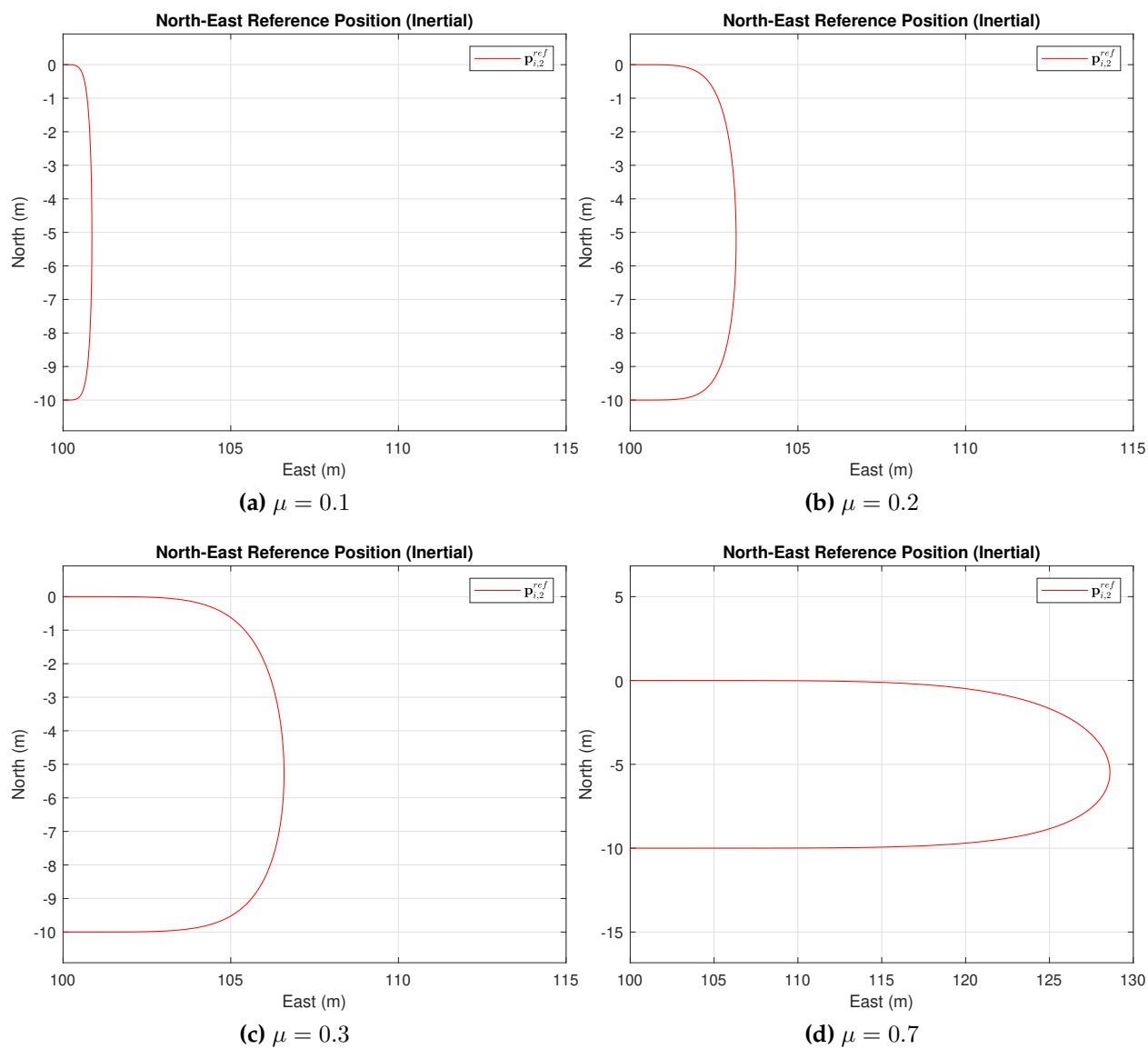


Figure 4.5: Second segment of the maneuver with 4 waypoints, at different turn ratios.

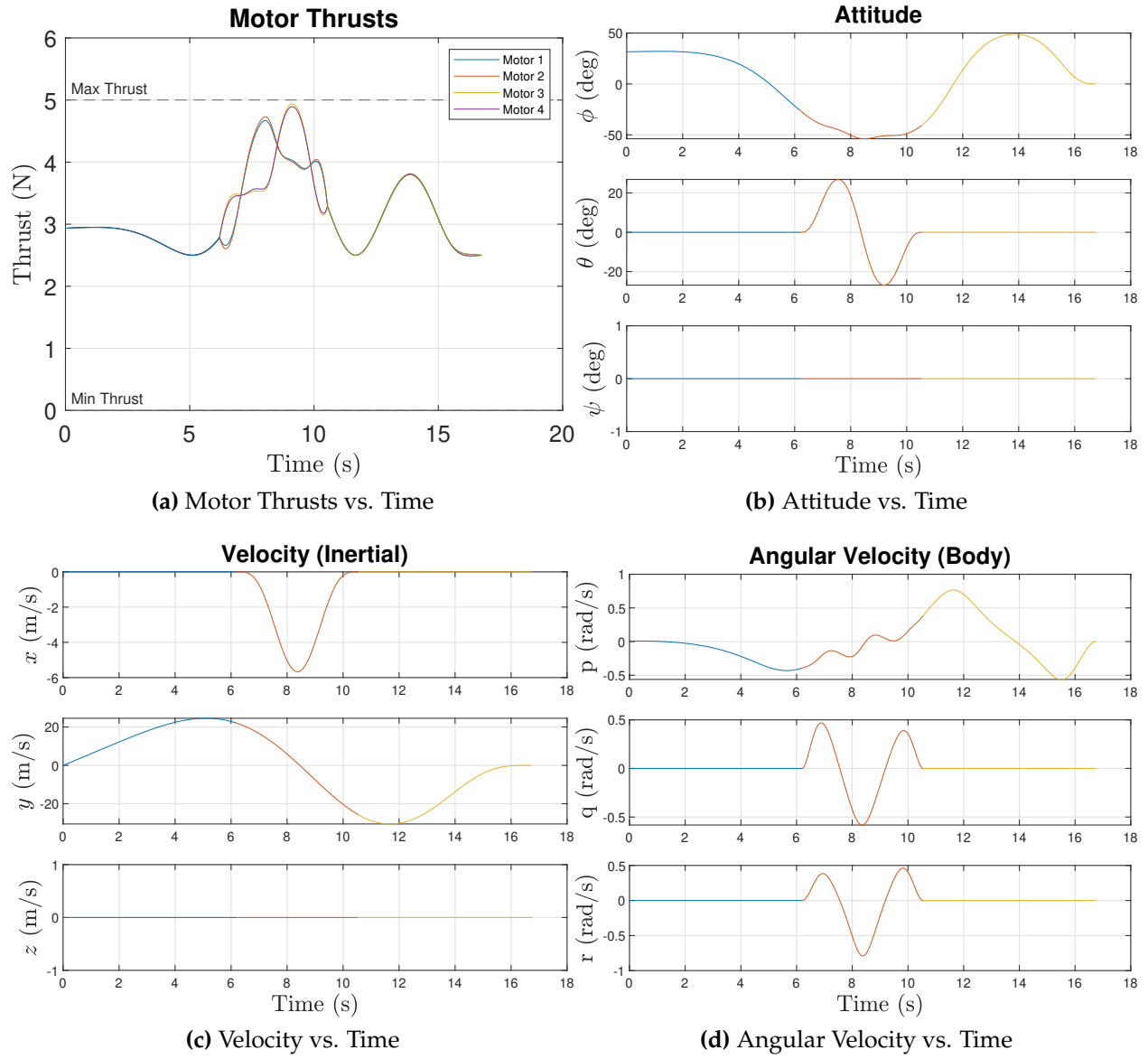


Figure 4.6: Plots obtained from differential flatness equations for the scaled trajectory with 4 waypoints, $\mu = 0.7$.

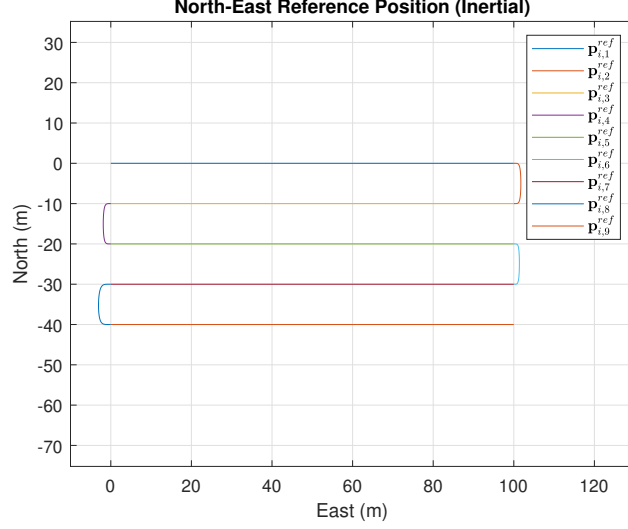


Figure 4.7: Position trajectory given $T_1 = T_3 = T_5 = T_7 = T_9 = 10s$ and $T_2 = T_4 = T_6 = T_8 = 2s$.

4.7 Second Trajectory Generation and Scaling Example

To further test the capabilities of minimum snap trajectory generation and time scaling, we expanded the maneuver evaluated in Sec. 4.5 to include 10 waypoints instead of 4. This new maneuver can be described as a continuation and repetition of that shown in Fig. 4.1 where the vehicle makes a turn at waypoint 4 to waypoint 5, then continues along a straight line towards waypoint 6, and so forth. The same assumptions made in the previous example were used here: the average velocity of the vehicle is kept at 10 m/s for estimating the segment times, and the yaw is set to zero throughout the maneuver. Durations for segments associated with a turn are set to 2 seconds (i.e., $T_2 = T_4 = T_6 = T_8 = 2s$). Altering this value will similarly affect the spatial properties of the turns, which will be investigated in Sec. 4.7.1.

Using these segment times, the QP problem was solved to obtain the position trajectory shown in Fig. 4.7. The segments corresponding to turns are shown in Fig. 4.8. In Fig. 4.8, we can see that the maximum turn displacement in the Easterly direction is between 1 to 3 m, with the largest seen in the eighth position segment at around 3 m, similar to what was obtained in the previous example for its second segment (shown in Fig. 4.2). We can

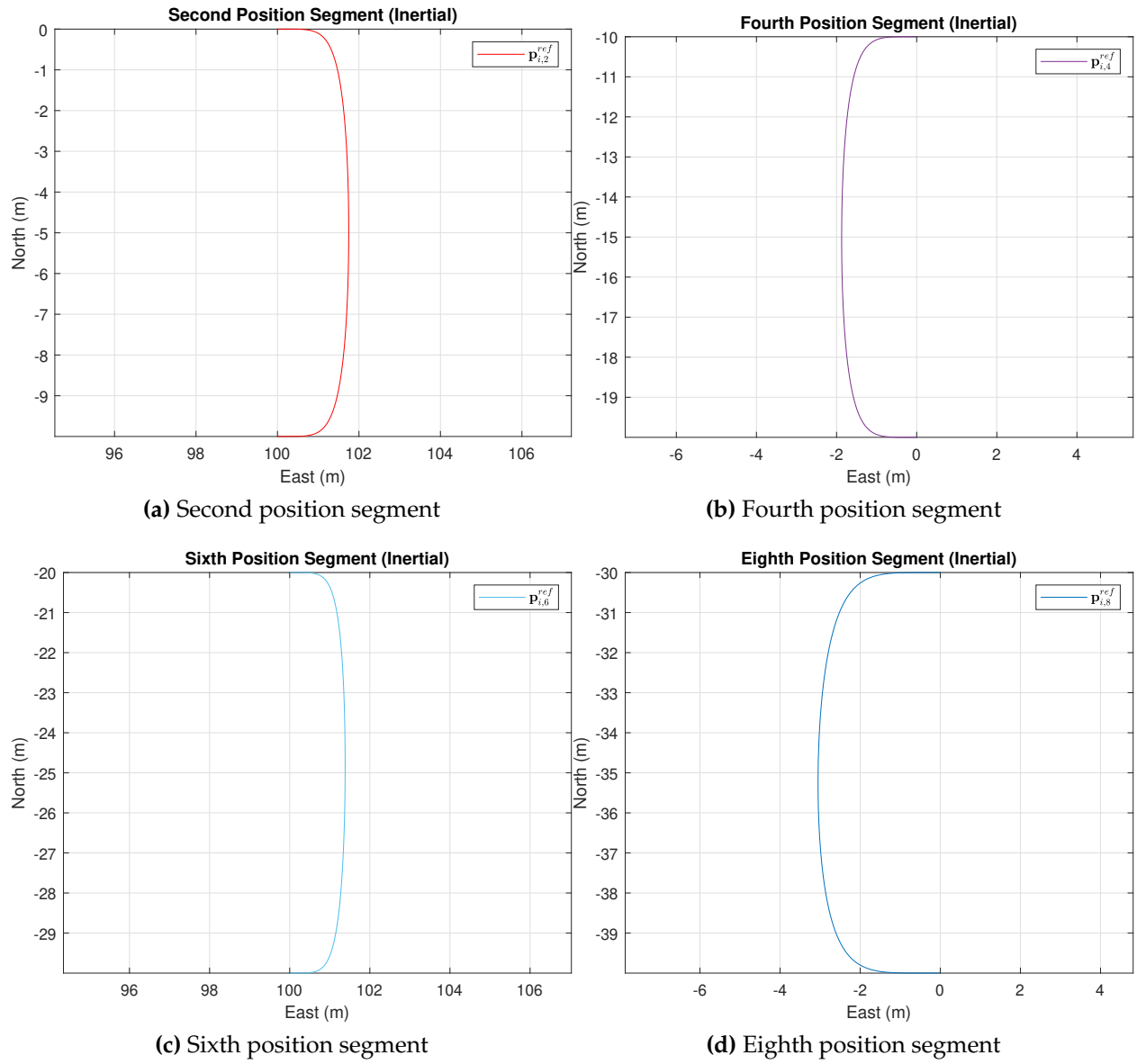


Figure 4.8: Position segments corresponding to turns in the maneuver with 10 waypoints.

Table 4.2: Turn ratios and total time for maneuver with 10 waypoints

μ	Total time (s)
0.1	129.6
0.2	75.4
0.3	58.9
0.4	52.8
0.5	49.7
0.6	48.1
0.7	47.58
0.8	48.38
0.9	49.02
1	50.4

now use differential flatness equations to plot our vehicle's states and inputs throughout this trajectory, as shown in Fig. 4.9. A maximum motor thrust constraint of 5 N will again be considered here and it is shown to be violated by our computed trajectory in this figure. Therefore, as before, iterative time scaling is needed to modify the trajectory and correct this issue. A scale of 1.3 was found to satisfy this condition, as shown in Fig. 4.10, which is the same scale obtained in the previous example with only 4 waypoints.

Comparing the states and inputs from both examples, the ones from the latest maneuver (shown in Fig. 4.9) appear to have very similar profiles to those in the previous example (shown in Fig. 4.3), but with additional repetitions. Some of the profiles are not exactly the same, as can be seen in the thrust profiles. Such differences are expected as more waypoints are added and the minimum snap trajectory optimization problem becomes more complex. Nevertheless, given the overall similarities, it is not surprising that the same scale was necessary to satisfy the actuator constraints.

4.7.1 Time Optimal Trajectory

We conducted the same analysis as in Sec. 4.6 to find the initial segment times corresponding to the trajectory needing the shortest total time. The same process of varying

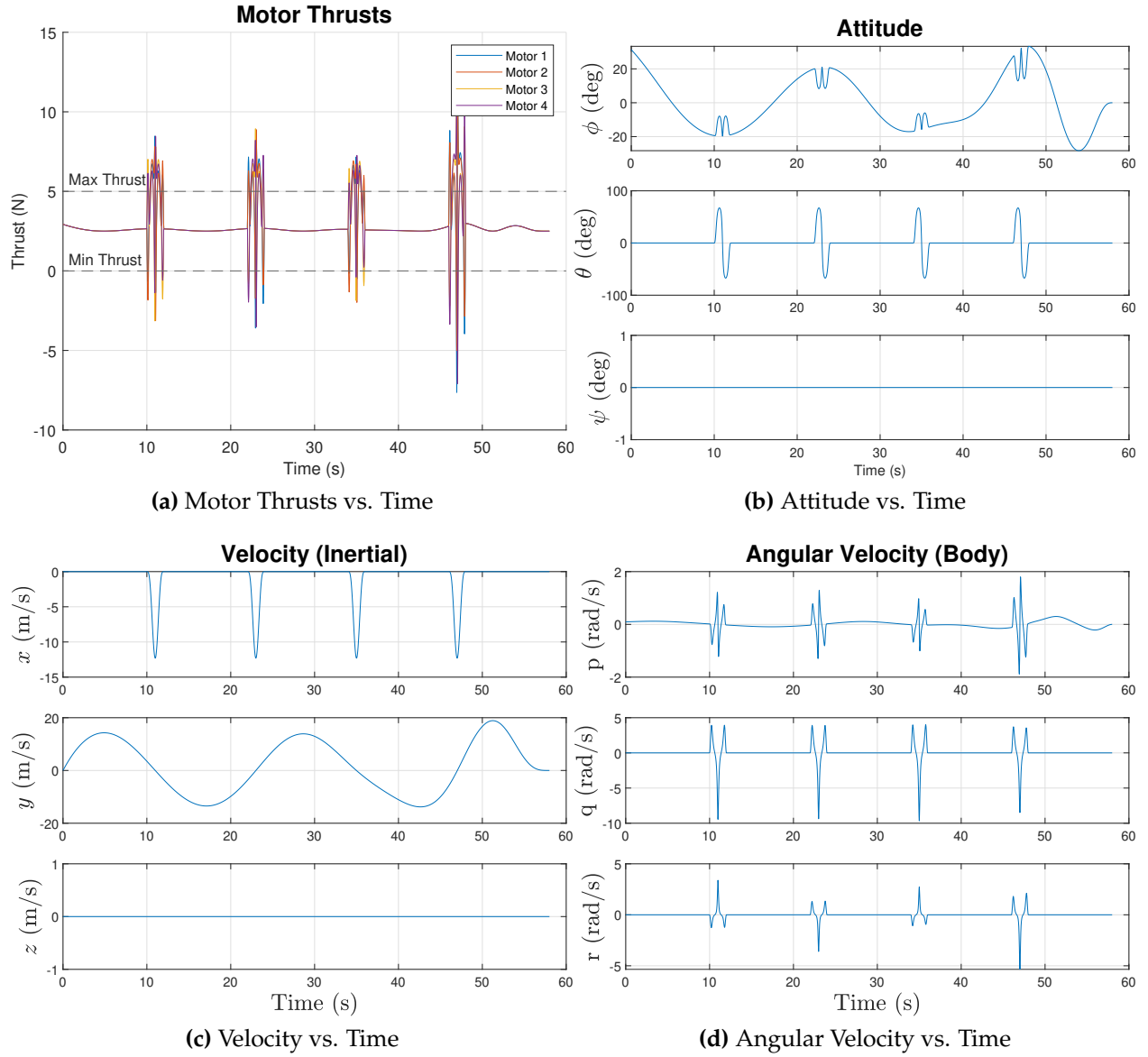


Figure 4.9: Plots obtained from differential flatness equations for the computed trajectory with 10 waypoints.

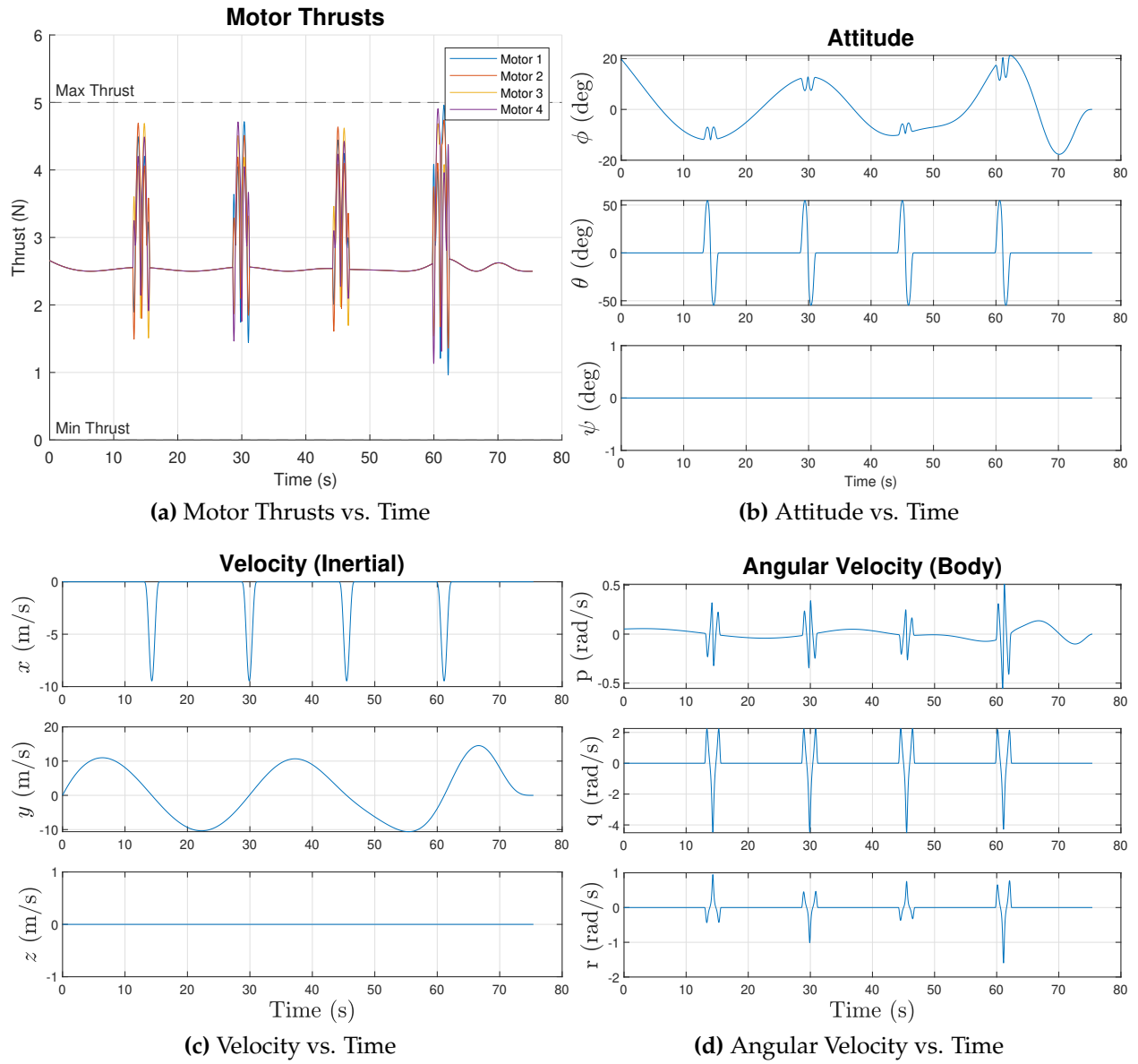


Figure 4.10: Plots obtained from differential flatness equations for the scaled trajectory with 10 waypoints, $s = 1.3$.

the turn ratios and applying time scaling was utilized, with results tabled in Table 4.2. From this table, we find that a turn ratio of 0.7 again yields the minimum total time. The shapes of the segments corresponding to turns with this ratio are shown in Fig. 4.11, and the states and inputs associated with the overall trajectory are shown in Fig. 4.12.

Comparing the turn displacements in Fig. 4.11 with those obtained in the previous scenario (shown in Fig. 4.5) at $\mu = 0.7$, we can see that the eighth segment in the current scenario and the second segment in the previous scenario have the same turn displacement of around 27 *m*. Both of these segments share the same property of being the final turn that the vehicle makes before its run to the last waypoint, at which the vehicle comes to a stop. Comparing the states and inputs in Fig. 4.12 with those obtained in the previous scenario (shown in Fig. 4.6), the maximum thrust appears to be generated at precisely these same segments in both scenarios. This suggests that by extending the lawnmower maneuver by adding more segments, we ‘shift’ the trajectory shape and profile of the three segments originally (shown in Fig. 4.1) to the final three of the extended version.

4.8 Analytical Time Scaling

As previously mentioned, finding the appropriate scale that satisfies all state and actuator constraints analytically can be difficult given the inherent nonlinearities in some of the differential flatness equations. Hence, an iterative approach was suggested to find the appropriate scale. There are, however, equations for certain states that do not have this issue, and there are others where assumptions need to be made in order to circumvent the nonlinearities; scales can be obtained analytically for constraints on the velocity, acceleration, total thrust, and attitude. This approach offers a far more accurate and computationally-efficient alternative to the iterative approach when only these constraints are being considered. On the other hand, if an iterative approach is preferred, this method can be used to find an initial scale that serves as an excellent starting point for the iterations.

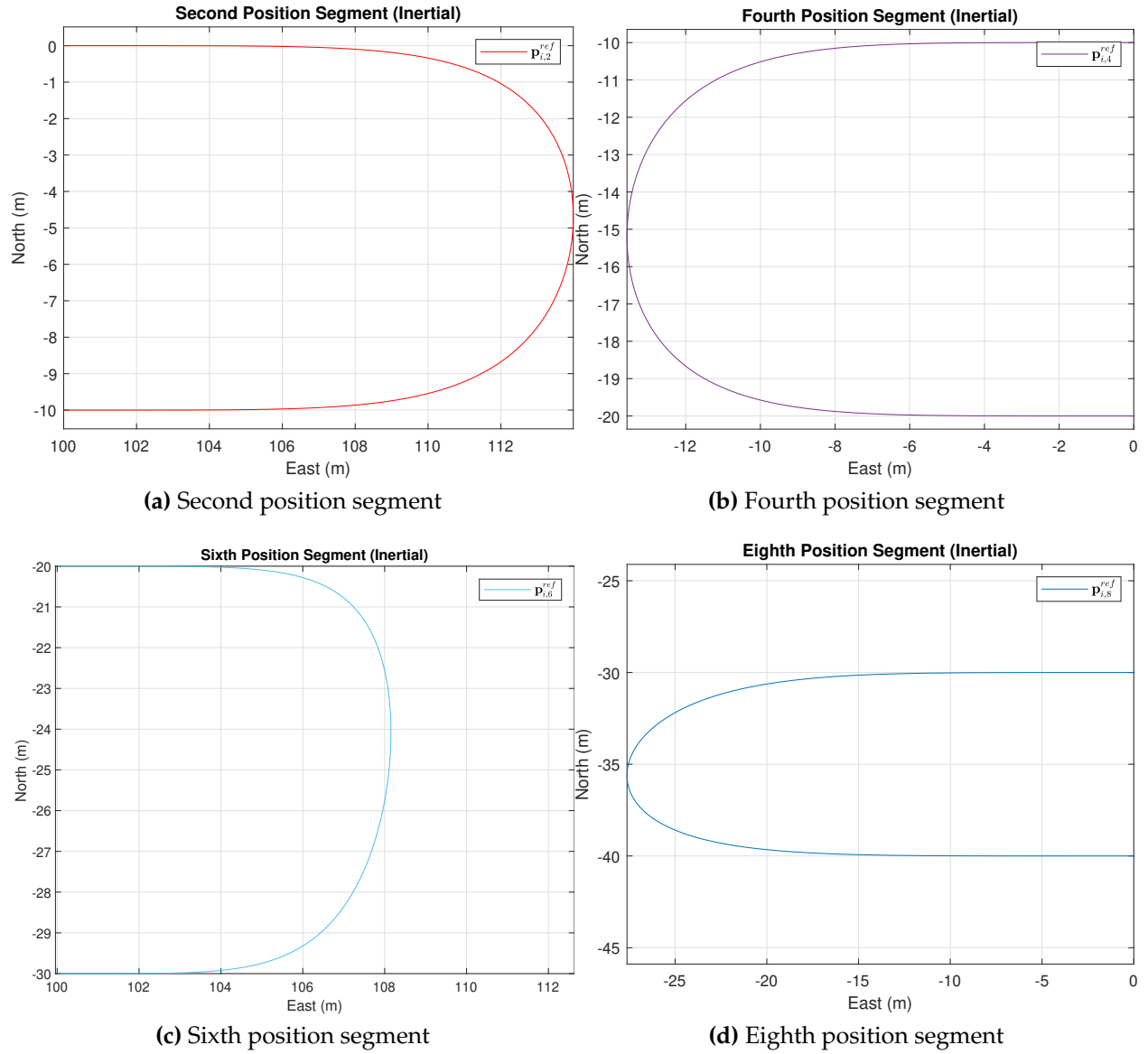


Figure 4.11: Position segments corresponding to turns in the maneuver with 10 waypoints, $\mu = 0.7$.

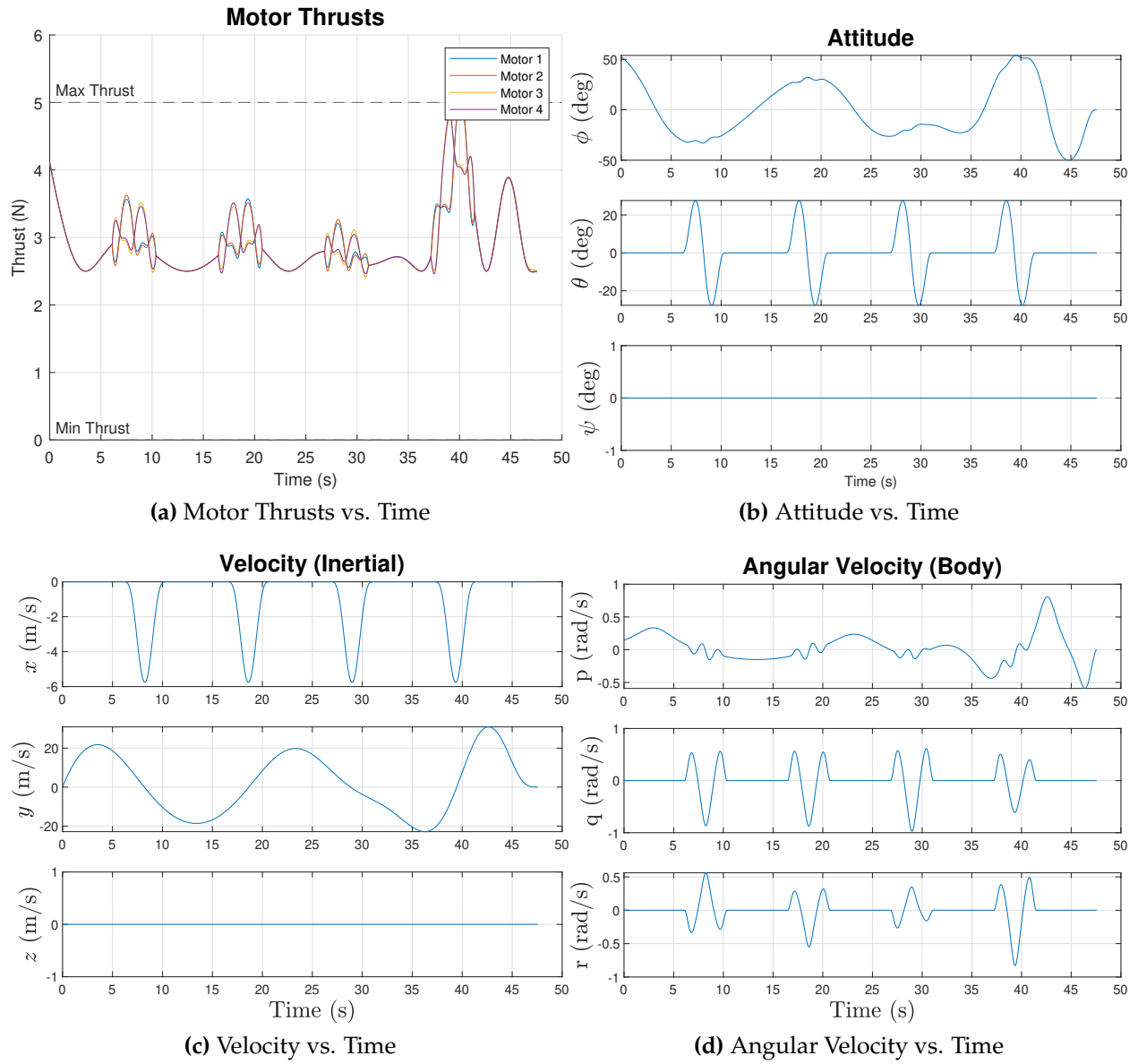


Figure 4.12: Plots obtained from differential flatness equations for the scaled trajectory with 10 waypoints, $\mu = 0.7$.

4.8.1 Velocity

Let us first consider a given velocity constraint along the inertial x -axis across all segments of the trajectory. To find the appropriate scale that satisfies this velocity constraint, we must find the following ratio as a function of s :

$$\frac{\dot{p}_{i,x}^{ref,scaled}(st)}{\dot{p}_{i,x}^{ref}(t)} \quad (4.39)$$

Given (4.35)–(4.37), it can be deduced that:

$$p_{i,x}^{ref,scaled}(st) = p_{i,x}^{ref}(t) \quad (4.40)$$

Therefore:

$$\dot{p}_{i,x}^{ref}(t) = c_1 + 2c_2t + \dots + 9c_9t^8 \quad (4.41)$$

$$\dot{p}_{i,x}^{ref,scaled}(t) = \frac{c_1}{s} + \frac{2c_2t}{s^2} + \dots + \frac{9c_9t^8}{s^9} \quad (4.42)$$

$$\dot{p}_{i,x}^{ref,scaled}(st) = \frac{c_1}{s} + \frac{2c_2t}{s} + \dots + \frac{9c_9t^8}{s} \quad (4.43)$$

$$\frac{\dot{p}_{i,x}^{ref,scaled}(st)}{\dot{p}_{i,x}^{ref}(t)} = \frac{1}{s} \quad (4.44)$$

In (4.44), $\dot{p}_{i,x}^{ref,scaled}$ and $\dot{p}_{i,x}^{ref}$ refer to the post-scaled and original trajectories, respectively.

To find s , $\dot{p}_{i,x}^{ref,scaled}$ must be the new maximum velocity (i.e., the velocity constraint) and $\dot{p}_{i,x}^{ref}$ the maximum velocity originally. The relationship can be generalized to all three axes:

$$\dot{\mathbf{p}}_i^{ref,scaled}(st) = \frac{1}{s} \dot{\mathbf{p}}_i^{ref}(t) \quad (4.45)$$

It can also be shown that:

$$\|\dot{\mathbf{p}}_i^{ref,scaled}(st)\| = \frac{1}{s} \|\dot{\mathbf{p}}_i^{ref}(t)\| \quad (4.46)$$

With this, we can find the appropriate scale for a given norm velocity constraint as well.

4.8.2 Acceleration

The same steps for velocity can be used to find the appropriate scale for a given acceleration constraint:

$$\ddot{\mathbf{p}}_i^{ref,scaled}(st) = \frac{1}{s^2} \ddot{\mathbf{p}}_i^{ref}(t) \quad (4.47)$$

Similarly, it can be shown that:

$$\|\ddot{\mathbf{p}}_i^{ref,scaled}(st)\| = \frac{1}{s^2} \|\ddot{\mathbf{p}}_i^{ref}(t)\| \quad (4.48)$$

4.8.3 Total Thrust

For a given total thrust constraint, an analytical solution can be found if we assume that $\ddot{p}_{i,z}^{ref}$ is zero. In other words, it only applies to maneuvers with a constant altitude (i.e., level-flight). Using (4.12):

$$T_{total}(t) = m \sqrt{\dot{p}_{i,x}^{ref}(t)^2 + \dot{p}_{i,y}^{ref}(t)^2 + g^2} \quad (4.49)$$

$$T_{total}^{scaled}(st) = m \sqrt{\dot{p}_{i,x}^{ref,scaled}(st)^2 + \dot{p}_{i,y}^{ref,scaled}(st)^2 + g^2} \quad (4.50)$$

Given (4.47), we can find the relationship:

$$\frac{T_{total}^{scaled}(st)^2 - (mg)^2}{T_{total}(t)^2 - (mg)^2} = \frac{1}{s^4} \quad (4.51)$$

4.8.4 Attitude

Yaw angle

Applying the same time scaling method to yaw trajectories, as was done for position in (4.35)–(4.37), it can be deduced that:

$$\psi^{ref,scaled}(st) = \psi^{ref}(t) \quad (4.52)$$

Therefore, the yaw trajectory profile will remain the same regardless of the scale used.

Pitch angle

Similar to the total thrust, an analytical solution for the pitch could only be found for level-flight maneuvers. Using (4.13):

$$\tan \theta^d = \frac{\cos \psi^{ref} \ddot{p}_{i,x}^{ref} + \sin \psi^{ref} \ddot{p}_{i,y}^{ref}}{-g} \quad (4.53)$$

$$\tan \theta^{d,scaled} = \frac{\cos \psi^{ref,scaled} \ddot{p}_{i,x}^{ref,scaled} + \sin \psi^{ref,scaled} \ddot{p}_{i,y}^{ref,scaled}}{-g} \quad (4.54)$$

The function variables (st) and (t) are removed in these equations for compactness, but it should be noted that they are associated with the scaled and non-scaled trajectories, respectively (e.g., $\theta^{d,scaled}$ is really $\theta^{d,scaled}(st)$). Given (4.47) and (4.52), we can find the relationship:

$$\frac{\tan \theta^{d,scaled}}{\tan \theta^d} = \frac{1}{s^2} \quad (4.55)$$

Roll angle

The same procedure used for the pitch angle can be followed for the roll angle:

$$s^4 = \frac{\sin^2 \phi^d}{\sin^2 \phi^{d.scaled}} \frac{(\ddot{p}_{i,x}^{ref})^2 + (\ddot{p}_{i,y}^{ref})^2 + g^2}{g^2} - \frac{(\ddot{p}_{i,x}^{ref})^2 + (\ddot{p}_{i,y}^{ref})^2}{g^2} \quad (4.56)$$

This solution is much more complex than the others but is solvable given that $\ddot{p}_{i,x}^{ref}$ and $\ddot{p}_{i,y}^{ref}$ are known accelerations at the maximum roll angle of the original trajectory.

4.9 Analytical Scaling Example

Let us use the same initial trajectory generated in Sec. 4.5 of this thesis and apply analytical time scaling given some constraints. The differential flatness plots for the initial trajectory are shown in Fig. 4.3. In this figure, we can see that the maximum pitch angle is around 67° . Let us prescribe a maximum pitch constraint of 60° and analytically find the appropriate scale to meet this constraint. Since the trajectory includes level-flight, we can use (4.55):

$$s = \sqrt{\frac{\tan 67^\circ}{\tan 60^\circ}} = 1.166 \quad (4.57)$$

Using this scale, we can see that the new maximum pitch is now at precisely 60° in Fig. 4.13.

Let us now consider a case where we also have maximum norm velocity and total thrust constraints in addition to the pitch constraint. In this case, three scales need to be computed, each with respect to their corresponding constraint. Afterwards, the largest of these scales is selected as the final solution as it would guarantee that all constraints are satisfied. For example, let us assume a maximum norm velocity and total thrust constraint of 16 m/s and 20 N , respectively. The maximum norm velocity and total thrust of the original trajectory is 18.9 m/s and 26.73 N . For the norm velocity constraint, we use (4.46)

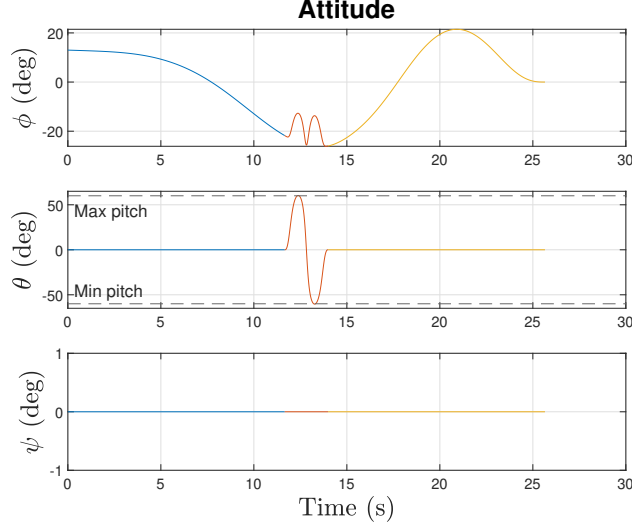


Figure 4.13: Attitude vs. Time plot for the scaled trajectory with constraints on the pitch angle.

to find the scale:

$$s = \frac{18.9}{16} = 1.181 \quad (4.58)$$

For the total thrust constraint, we use (4.51):

$$s = \sqrt[4]{\frac{26.73^2 - (10)^2}{20^2 - (10)^2}} = 1.196 \quad (4.59)$$

Of the three computed scales, the largest is found to be 1.196. Using this scale, we can see that constraints for the attitude, norm velocity, and total thrust are satisfied in Fig. 4.14. However, Fig. 4.14a indicates that the individual motor thrusts are still being violated despite the total thrust meeting its constraint. This is predictable, since it was (iteratively) found in Sec. 4.5 that a scale of 1.3 was needed to remain within the individual thruster limits. Since an analytical solution for the individual thrusts could not be deduced, avoiding this violation was not guaranteed, thus further manual iteration is needed beyond this point. Nevertheless, this approach demonstrates an efficient way of finding the appropriate scale and, if it fails to do so, serves as an excellent starting point for an iterative approach.

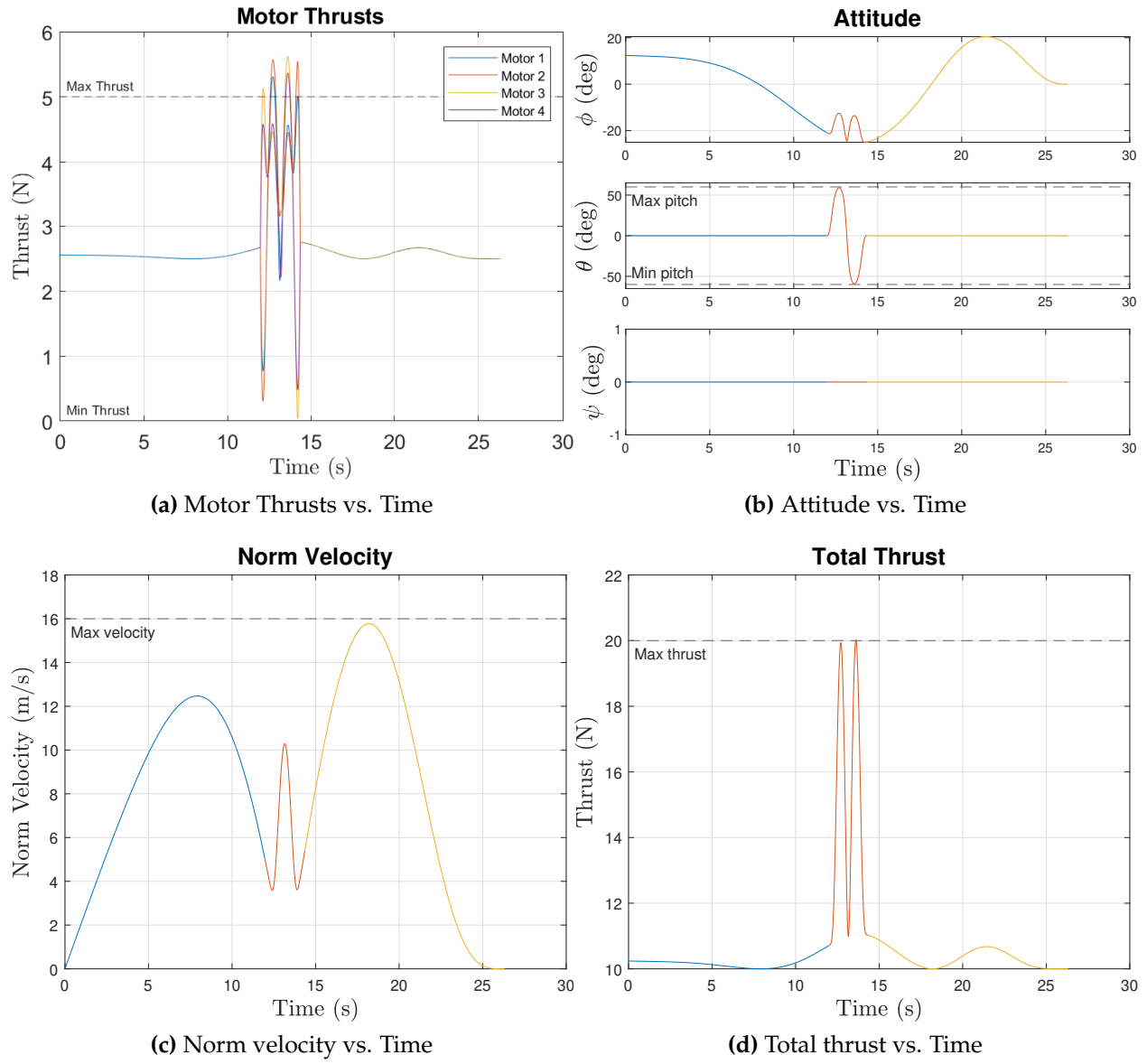


Figure 4.14: Plots obtained for the scaled trajectory with constraints shown.

Chapter 5

Conclusions and Future Work

5.1 Summary of Research

This thesis presents a conventional control system, based on a cascaded control structure, to mitigate the adverse effects of actuator saturation caused by external disturbances, such as wind gusts. Modifications to the component sub-controllers, including the mixer, the attitude controller, and the position controller are presented and evaluated. This thesis also investigates the use of a waypoints-based motion planner for generating trajectories that ‘avoid’ saturation, as opposed to ‘handling’ it in the controllers and mixer.

The first research objective was to assemble models of the quadrotor’s dynamics and kinematics, the thruster dynamics, the maneuver to be executed, and the wind disturbance, to serve as the building blocks of the simulation environment. The maneuver was designed to just reach saturation without any wind disturbance, and this disturbance was modeled as a lateral force that exacerbates saturation. A conventional controller was incorporated into the simulation to demonstrate its behavior under these conditions. It was shown that the conventional control system had no issues with trajectory-tracking when there are no actuator constraints (i.e, there is no actuator saturation). However, when

actuator constraints were imposed, the conventional control system was unable to cope with actuator saturation, and so the vehicle became unstable.

The next research objective was to propose and investigate modifications to each conventional control subsystem in order to mitigate the adverse effects of saturation. A constrained weighted least squares approach was proposed for the mixer to enable relative control effort prioritization. With this method, we were able to improve vehicle stability by prioritizing reduced attitude control over total thrust and yaw control. Attempts to improve altitude control by prioritizing total thrust control over yaw control led to the vehicle becoming unstable. It was discovered that this was due to the coupling of the yaw and reduced attitude error dynamics in the conventional attitude controller. Improved altitude control was desired to reduce the risk of the vehicle crashing into the ground. A decoupled attitude controller was proposed to decouple the error dynamics. With the vehicle now stable, another weighted least squares approach was proposed for the position controller to further improve altitude control. This method allowed us to prioritize the control effort corresponding to altitude control, which would consequentially lead to a slight degradation in horizontal position control. Overall, these modifications were found to significantly improve the trajectory tracking performance and stability of the vehicle when compared to the conventional control system.

The final research objective was to implement a motion planner that generates trajectories for a ‘lawnmower maneuver’. The motion planner presented, which is based on a waypoints-based minimum-snap approach, was shown to be highly effective at producing dynamically feasible trajectories in a computationally-efficient manner. However, this method does not take actuator or state constraints into consideration, and thus avoiding constraint violations is not guaranteed. To address this, differential flatness properties were used to evaluate these trajectories to determine whether actuator and state constraints were violated. Time scaling was then used to adjust the trajectory to meet the constraints. The appropriate scale needed for time scaling can be obtained analytically or iteratively. It was found that analytical solutions exist for constraints on the vehicle’s

velocity, acceleration, total thrust, and attitude. These solutions serve as excellent starting points for states and inputs that require an iterative approach.

5.2 Suggestions for Future Work

The research conducted in this work opens many avenues for further experimentation. For example, we only considered controllers that were based on PD controllers. An integral term may be used to mitigate any steady-state errors seen in the position and attitude control. In this scenario, given that we are considering cases where there is actuator saturation, steady-state errors would continue to grow during periods of saturation, leading to integrator windup. Thus, anti-windup techniques would need to be investigated and incorporated alongside PID controllers.

Additionally, it would be worth investigating if the proposed modifications to PD controllers in this work could be tailored for other types of controllers. For example, if Model Predictive Control (MPC), sliding mode control, backstepping control, etc. were used for position and attitude control, would it be possible to use constrained control allocation approaches to improve their handling of actuator constraints?

It was noted in Sec. 3.3 that, interestingly, the optimal position controller can optionally be used to output yaw setpoints. This property, of course, assumes that the flight operator is indifferent to the vehicle's yaw during flight, which was not the case in the present work. In [25], this property was investigated where the authors saw a slight improvement in trajectory-tracking performance when compared to the case where the yaw setpoint was a constant value. It would be interesting to conduct this same experiment to verify these observations alongside the use of a decoupled attitude controller and optimal mixer. Further development of the optimal position controller would also be needed to determine whether it can be implemented in real-time on flight hardware.

For time scaling, we were only able to obtain analytical solutions for a few vehicle states and inputs. States and inputs such as the angular velocity and individual motor thrusts, have highly nonlinear differential flatness equations, and more work would be needed to find their corresponding analytical solutions, if possible.

This thesis only focuses on the quadrotor platform despite the fact that constrained control allocation methods may be needed on larger vehicles, such as hexarotors (6 motors) and octorotors (8 motors). With more actuators, the optimization problems in the optimal mixer and optimal position controller would become more computationally expensive to solve. How this increase in computational complexity scales with the number of rotors would be valuable research.

Finally, on the subject of practicality, we have only tested our proposed methods in simulation. It would be imperative to perform hardware-in-the-loop simulations and real flight testing to determine the practicality and effectiveness of the methods proposed here.

Bibliography

- [1] L. F. Luque-Vega, B. Castillo-Toledo, A. Loukianov, and L. E. Gonzalez-Jimenez, "Power line inspection via an unmanned aerial system based on the quadrotor helicopter," in *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, pp. 393–397, 2014.
- [2] S. Zhang, H. Wang, S. He, C. Zhang, and J. Liu, "An autonomous air-ground cooperative field surveillance system with quadrotor uav and unmanned atv robots," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1527–1532, 2018.
- [3] B. Renardi, E. Khosasi, Y. Y. Nazaruddin, and E. Juliastuti, "Using multi-quadrotor system for effective road mapping," in *2019 6th International Conference on Electric Vehicular Technology (ICEVT)*, pp. 246–252, 2019.
- [4] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 2, pp. 476–482, April 2017.
- [5] M. Wilfried Mueller, "Multicopter attitude control for recovery from large disturbances," *arXiv e-prints*, Feb. 2018.
- [6] E. Smeur, D. Höppener, and C. De Wagter, "Prioritized Control Allocation for Quadrotors Subject to Saturation," in *International Micro Air Vehicle Conference and Flight Competition 2017* (H. d. P. J.-M. Moschetta, G. Hattenberger, ed.), pp. 37–43, Sep. 2017.

- [7] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on $SE(3)$," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, Dec. 2010.
- [8] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2451–2456 vol.3, Sep. 2004.
- [9] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2247–2252, April 2005.
- [10] D. Brescianini and R. D'Andrea, "Tilt-prioritized quadrocopter attitude control," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2018.
- [11] T. Johansen and T. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, p. 1087–1103, May 2013.
- [12] M. Kirchengast, M. Steinberger, and M. Horn, "Control allocation under actuator saturation: An experimental evaluation," *IFAC-PapersOnLine*, vol. 51, no. 25, pp. 48 – 54, 2018.
- [13] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 703–711, 2002.
- [14] J. C. Monteiro, F. Lizarralde, and Liu Hsu, "Optimal control allocation of quadrotor uavs subject to actuator constraints," in *2016 American Control Conference (ACC)*, pp. 500–505, 2016.
- [15] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.
- [16] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on $SO(3)$," in *2015 IEEE Conference on Control Applications (CCA)*, pp. 1160–1166, Sep. 2015.

- [17] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear PID control of a quadrotor UAV on SE(3)," in *2013 European Control Conference (ECC)*, pp. 3845–3850, July 2013.
- [18] D. Brescianini, M. Hehn, and R. D’Andrea, "Nonlinear quadrocopter attitude control. technical report," tech. rep., Eidgenössische Technische Hochschule Zürich, Zürich, 2013.
- [19] N. Chaturvedi, S. Amit, and N. McClamroch, "Rigid-body attitude control," *Control Systems, IEEE*, vol. 31, pp. 30 – 51, Aug. 2011.
- [20] K. Gamagedara, M. Bisheban, E. Kaufman, and T. Lee, "Geometric controls of a quadrotor uav with decoupled yaw control," in *2019 American Control Conference (ACC)*, pp. 3285–3290, July 2019.
- [21] D. Kooijman, A. P. Schoellig, and D. Antunes, "Trajectory tracking for quadrotors with attitude control on $s^2 \times s^1$," *ArXiv*, vol. abs/1906.06926, 2019.
- [22] Y. Aoki, Y. Asano, A. Honda, N. Motooka, and T. Ohtsuka, "Nonlinear model predictive control of position and attitude in a hexacopter with three failed rotors," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 228 – 233, 2018.
- [23] M. Rosaldo-Serrano and E. Aranda-Bricaire, "Trajectory tracking for a commercial quadrotor via time-varying backstepping," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 532 – 536, 2018.
- [24] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, May 2011.
- [25] H. Zaki, M. Unel, and Y. Yildiz, "Trajectory control of a quadrotor using a control allocation approach," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 533–539, June 2017.
- [26] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*, pp. 649–666, April 2016.

- [27] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*, pp. 3936–3943, 2019.
- [28] M. Cutler and J. How, "Actuator constrained trajectory generation and control for variable-pitch quadrotors," *AIAA Guidance, Navigation, and Control Conference 2012*, Sep. 2012.
- [29] G. Yu, D. Cabecinhas, R. Cunha, and C. Silvestre, "Quadrotor trajectory generation and tracking for aggressive maneuvers with attitude constraints," *IFAC Symposium on Automatic Control in Aerospace ACA 2019*, vol. 52, pp. 55–60, Jan. 2019.
- [30] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1872–1878, 2015.
- [31] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, p. 620–626, April 2018.
- [32] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *2013 European Control Conference (ECC)*, pp. 3864–3869, July 2013.
- [33] W. Khan and M. Nahon, "Toward an accurate physics-based uav thruster model," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1269–1279, 2013.
- [34] W. Jothiraj, C. Miles, E. Bulka, I. Sharf, and M. Nahon, "Enabling bidirectional thrust for aggressive and inverted quadrotor flight," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 534–541, June 2019.
- [35] W. Jothiraj, I. Sharf, and M. Nahon, "Control allocation of bidirectional thrust quadrotor subject to actuator constraints," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 932–938, Sep. 2020.

- [36] P. Pounds, R. Mahony, J. Gresham, P. Corke, and J. M. Roberts, "Towards dynamically-favourable quad-rotor aerial robots," in *Proceedings of the 2004 Australasian Conference on Robotics & Automation*, Australian Robotics & Automation Association, 2004.
- [37] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," *2006 Australasian Conference on Robotics and Automation*, Dec. 2006.
- [38] N. K. Tran, E. Bulka, and M. Nahon, "Quadrotor control in a wind field," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 320–328, 2015.
- [39] N. Sydney, B. Smyth, and D. A. Paley, "Dynamic control of autonomous quadrotor flight in an estimated wind field," in *52nd IEEE Conference on Decision and Control*, pp. 3609–3616, 2013.
- [40] R. Chiappinelli, M. Nahon, and J. Apkarian, "The simulator-in-hardware: a low cost and hard real-time hardware-in-the-loop simulator for flying vehicles," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1766–1772, Sep. 2020.
- [41] N. Cao and A. F. Lynch, "Inner-outer loop control for quadrotor uavs with input and state constraints," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1797–1804, 2016.
- [42] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, March 2018.
- [43] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, p. 1258–1276, Dec. 2013.
- [44] J. Fan, *Optimal Path Planning and Control of Quadrotor Unmanned Aerial Vehicle for Area Coverage*. PhD thesis, The University of Toledo, Jan. 2014.
- [45] C. Berger, M. Wzorek, J. Kvarnström, G. Conte, P. Doherty, and A. Eriksson, "Area coverage with heterogeneous uavs using scan patterns," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 342–349, 2016.

Appendix A

Weight Selection

To better understand the behaviour of the mixer, independently of the controller, we resorted to open-loop testing of the mixer block, by itself. This consisted of inputting time histories of desired forces and moments data into the optimal mixer, by itself, with a specific weighting matrix, and evaluating its output. The output would be time histories of desired individual thrust values which can then be used in (2.7) and (2.9) to find the actual net forces and moments being applied on the quadrotor. The error between the desired and actual forces and moments were then evaluated to accomplish two goals: (1) to establish a relationship between the weighting of elements with different units, as discussed in Sec. 3.1, and (2) to determine the magnitude of the weights needed to achieve a certain prioritization scheme, also noted in Sec. 3.1.

To reiterate the issue addressed by the first goal: there are elements with different units used in (3.3): one associated with a force, and three others with moments. These different weights will not have the same impact on their respective control efforts, and so some form of correspondence should be established between them. This is a commonly-encountered problem in any multi-objective optimization in which the component objectives have different units. We can make certain assumptions to find an approximate rela-

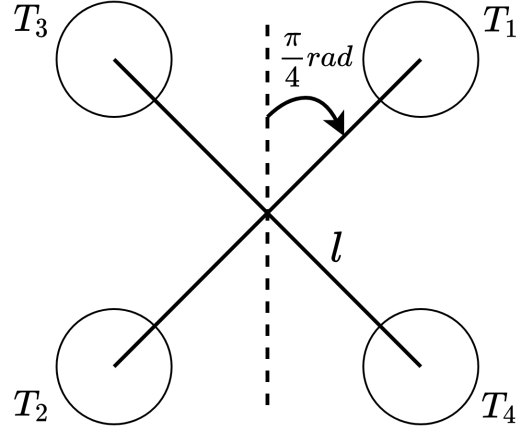


Figure A.1: Top view of the quadrotor model shown in Fig. 2.1.

tionship between the weights as a starting point, and then performing open-loop tests for verification.

One way to determine an ‘equivalence’ between force and moment magnitudes is to consider the moment generated by a given force applied on the quadrotor. We can consider a quadrotor in a configuration that maximizes the moments generated by thruster forces, as shown in Fig. A.1. It can be shown that, in this configuration, with maximum thrust on thrusters 2 & 3, and zero thrust on 1 & 4, we will obtain the maximum moment about the axis shown. Under these conditions, we find that the ratio of thruster force to thruster moment is:

$$\frac{2 * T_{max}}{2 * T_{max} \frac{l\sqrt{2}}{2}} = \frac{2}{l * \sqrt{2}} \quad (\text{A.1})$$

For an arm length of $l = 0.235 \text{ m}$, as used in the present work, this yields a force-to-moment ratio of 6.02 m^{-1} . That is, forces of a given magnitude will result in moments that are numerically about six times smaller. Therefore, in order to weight forces and moments ‘equivalently’, a weight that is six times higher should be used in the mixer for moments than for forces. Since the above analysis is approximate, at best, we chose to consider a weight 10 times smaller on the force as on the three moments to be equivalent, as a starting point.

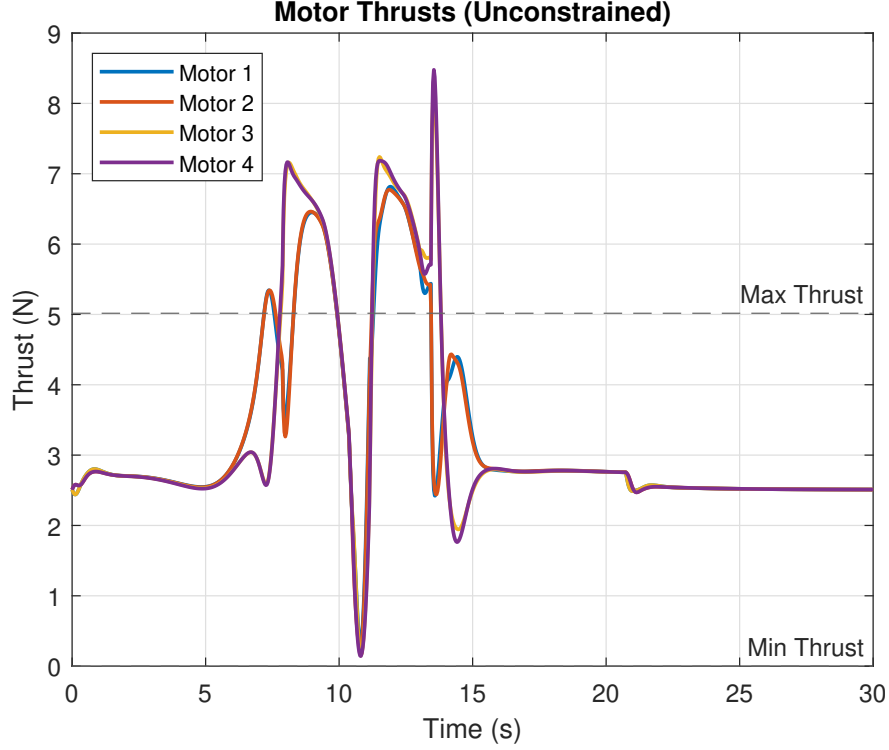


Figure A.2: Motor thrusts of all four motors vs. Time, using the conventional approach used in Fig. 2.3 without actuator constraints. Wind speed = 9.22 m/s .

The desired forced and moments data used for open-loop testing was obtained from a simulation similar to the one conducted in Sec. 2.2.4, without actuator constraints, but with a slightly stronger wind disturbance (side force magnitude = $\frac{3mg}{4}$, wind speed = 9.22 m/s). The stronger wind speed was used to generate more severe actuator saturation, and so more data could be obtained from the optimal mixer for evaluation. The resulting individual motor thrusts from this simulation are plotted in Fig. A.2.

With an approximated force-to-moment ratio of 10 m^{-1} and a weight of $1000 \text{ N}^{-1}\text{m}^{-1}$ on the three moments, the weighting matrix consisting of ‘equivalent’ weights is formulated as: $\mathbf{W}_m = \text{diag}\{100, 1000, 1000, 1000\}$. Open-loop testing with this matrix shows significant error between the desired and actual values of all four control efforts suggesting that no single control effort is being prioritized, as should be expected given the ‘equivalent’ weights. It should be noted that the open-loop test results are highly susceptible to the maneuver being executed by the vehicle, and so any conclusions based on these re-

sults cannot be generalized. Nonetheless, given the maneuver used in the present work, a ratio of 10 m^{-1} was found to be adequate in obtaining a weighting matrix of ‘equivalent’ weights.

To determine the magnitude of the weights needed to achieve certain prioritization schemes, noted earlier as the second goal of open-loop testing, a trial-and-error process was used whereby the weights were varied until the desired level of prioritization was achieved. For example, the process of finding the weighting matrix which prioritizes roll and pitch control first, then yaw control, and finally total thrust control is as follows: the weights corresponding to the control efforts for roll and pitch control, $m_{b,x}^{des}$ and $m_{b,y}^{des}$, were set at $1000 \text{ N}^{-1}\text{m}^{-1}$. The weights corresponding to the control efforts for yaw and total thrust control, $m_{b,z}^{des}$ and T_{total}^{body} , were varied during open-loop testing until the error profile resembled the desired prioritization scheme. This process yielded the following weighting matrix: $\mathbf{W}_m = \text{diag}\{1, 1000, 1000, 100\}$. Closed-loop simulations with this matrix verified its effectiveness in achieving this prioritization scheme. Using this same process, we obtained the weighting matrix which prioritizes roll and pitch control first, then total thrust control, and finally yaw control, as: $\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$. It is acknowledged that there is some subjectivity in this process, when evaluating desired outcomes. Nevertheless this process was found to be reasonably systematic in achieving the desired goals.

Appendix B

Rate Constraints

Thus far we have only considered minimum and maximum thrust constraints of the actuators. There exists, however, limitations on the rate of change of a motor's rotational speed as well, called a rate constraint. In the optimal mixer, defined in (3.1), we do not consider these rate constraints. This leads to the question as to whether or not the performance of the optimal mixer (and, therefore, the trajectory-tracking performance of the vehicle) is realistic since it does not restrict the rate at which thrust can vary.

A motor's maximum possible rate of change of speed is referred to as its slew rate, and so the instantaneous maximum allowable change in its speed can be formulated, as follows:

$$\Delta\Omega_{max} = \dot{\Omega}_{slew} * t_s \quad (\text{B.1})$$

where $\Delta\Omega_{max}$ is the maximum allowable change in the motor's rotational speed in one time step, $\dot{\Omega}_{slew}$ is the slew rate, and t_s is the sampling time. The sampling time may correspond to the sampling time used on the vehicle's flight computer, or the step size used in simulations. With this relationship, we can formulate the rate constraints, as follows:

$$\Omega_{i,k}^{des} \leq \Omega_{i,k-1}^{des} + \Delta\Omega_{max} \quad (\text{B.2})$$

$$\Omega_{i,k}^{des} \geq \Omega_{i,k-1}^{des} - \Delta\Omega_{max} \quad (\text{B.3})$$

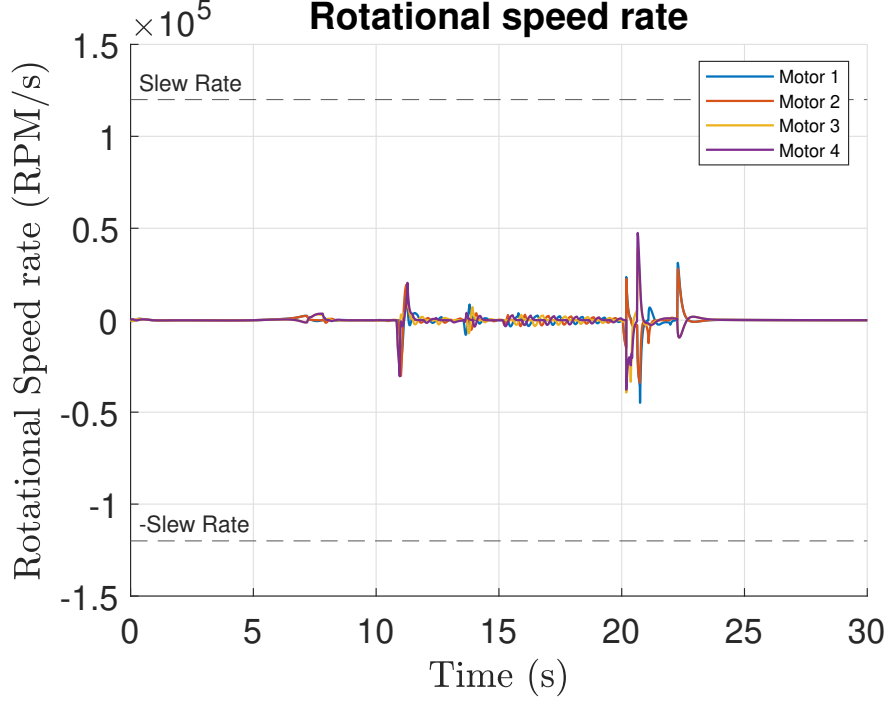


Figure B.1: Motor rotational speed rates using conventional position controller, decoupled attitude controller, optimal mixer ($\mathbf{W}_m = \text{diag}\{10, 1000, 1000, 1\}$), side force magnitude = mg , wind speed = 10.6 m/s .

where $\Omega_{i,k}^{des}$ refers to the desired speed of the i^{th} motor at a particular time step, k .

The slew rate of the motors on the Spiri quadrotor, the platform used in the present work, was found to be around $120,000 \text{ RPM/s}$ in [35]. Plotting the motor rotational speed rates obtained from the simulation conducted in Sec. 3.2 (with plots shown in Fig. 3.4) in Fig. B.1 shows us that this slew rate is never reached by the motors for the maneuver considered in this work. For the motors to reach this slew rate, the vehicle must be on the verge of, or has reached, instability. This can be inferred from the erratic motor speeds shown in Fig. 2.5 in which the vehicle goes unstable over time using the conventional control approach. In this case, the use of rate constraints in the optimal mixer may be considered as a potential solution to improve stability and control. However, it can be reasoned that if the vehicle experiences instability with only minimum and maximum thrust considerations in the optimal mixer, then the addition of rate constraints will excessively restrict their thrust variation and worsen stability and control. This was verified

in simulation and is supported by the findings in [35]. Based on this, we concluded that rate constraints should not be used in the optimal mixer for the maneuver considered in this work.

For the sake of completeness, the application of the rate constraints in the optimal mixer is described here, as follows: the constraints of the optimal mixer in (3.1) are defined in terms of thrust, and so the rate constraints in (B.2) and (B.3) must be converted to their thrust equivalent, using:

$$T_{i,k}^{des} \leq T_{i,k-1}^{des} + \Delta T_{max} \quad (\text{B.4})$$

$$T_{i,k}^{des} \geq T_{i,k-1}^{des} - \Delta T_{max} \quad (\text{B.5})$$

where $T_{i,k}^{des}$ refers to the desired thrust of the i^{th} motor at a particular time step, k . The instantaneous maximum possible change in the motor's thrust is denoted by ΔT_{max} . Using (2.8) we can find ΔT_{max} , as follows:

$$T_{i,k-1}^{des} + \Delta T_{max} = k_t(\Omega_{i,k-1}^{des} + \Delta\Omega_{max})^2 \quad (\text{B.6})$$

$$\Delta T_{max} = k_t(2\Delta\Omega_{max} * \Omega_{i,k-1}^{des} + \Delta\Omega_{max}^2) \quad (\text{B.7})$$

where $\Delta\Omega_{max}$ is given by (B.1). With these new constraints the optimal mixer is defined, as follows:

$$\begin{aligned} \min : J &= \|\mathbf{W}_m(\mathbf{b} - \mathbf{\Gamma}\mathbf{x})\|^2 \\ \text{s.t. : } \begin{bmatrix} T_{i,k}^{des} \\ -T_{i,k}^{des} \\ T_{i,k}^{des} \\ -T_{i,k}^{des} \end{bmatrix} &\leq \begin{bmatrix} T_{max} \\ -T_{min} \\ T_{i,k-1}^{des} + \Delta T_{max} \\ -T_{i,k-1}^{des} + \Delta T_{max} \end{bmatrix} \text{ for } i = 1, \dots, 4 \end{aligned} \quad (\text{B.8})$$

This new optimal mixer can also be formulated as a quadratic program and can therefore be solved efficiently.