

Article

Nonlinear Model Predictive Control for Unmanned Aerial Vehicles [†]

Pengkai Ru [‡] and Kamesh Subbarao ^{‡,*}

Department of Mechanical & Aerospace Engineering, University of Texas at Arlington, Arlington, TX 76019, USA; pengkai.ru@mavs.uta.edu

* Correspondence: subbarao@uta.edu; Tel.: +1-817-272-7467

[†] This paper is an extended version of our paper published in the Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA Aviation Forum, Dallas, TX, USA, 22–25 June 2015; Kamesh Subbarao, Carlos Tule and Pengkai Ru, Nonlinear Model Predictive Control Applied to Trajectory Tracking for Unmanned Aerial Vehicles, No. AIAA 2015-2857.

[‡] These authors contributed equally to this work.

Academic Editor: David Anderson

Received: 26 April 2017; Accepted: 12 June 2017; Published: 17 June 2017

Abstract: This paper discusses the derivation and implementation of a nonlinear model predictive control law for tracking reference trajectories and constrained control of a quadrotor platform. The approach uses the state-dependent coefficient form to capture the system nonlinearities into a pseudo-linear system matrix. The state-dependent coefficient form is derived following a rigorous analysis of aerial vehicle dynamics that systematically accounts for the peculiarities of such systems. The same state-dependent coefficient form is exploited for obtaining a nonlinear equivalent of the model predictive control. The nonlinear model predictive control law is derived by first transforming the continuous system into a sampled-data form and then using a sequential quadratic programming solver while accounting for input, output and state constraints. The boundedness of the tracking errors using the sampled-data implementation is shown explicitly. The performance of the nonlinear controller is illustrated through representative simulations showing the tracking of several aggressive reference trajectories with and without disturbances.

Keywords: nonlinear; model predictive control; constraints; trajectory tracking; stability

1. Introduction

A quadrotor helicopter platform (often just called a quadrotor) is an under-actuated helicopter with two pairs of rotors in a cross-configuration capable of spinning at different angular velocities in order to achieve translational and rotational motion. Rotor pair (1, 3) spins in one direction, while the pair (2, 4) spins in the opposite (see Figure 1).

The different motions the quadrotor can perform are: (a) vertical motion: simultaneous change in rotor speed; (b) roll motion: imbalance in the rotor speed of pair (2, 4); (c) pitch motion: imbalance in the rotor speed of pair (1, 3); (d) yaw motion: imbalance between all rotors.

Quadrotors have gained popularity as research platforms because of their simplicity of design and low cost of manufacturing. Because they are challenging vehicles to control, wherever operated in an indoor environment or in the open field, they serve as great platforms for research. They also possess many advantages over standard helicopters in terms of safety and efficiency at small sizes [1]. There are many applications for a quadrotor platform, both in the military and the civil sectors, which are summarized quite extensively in [2–6].

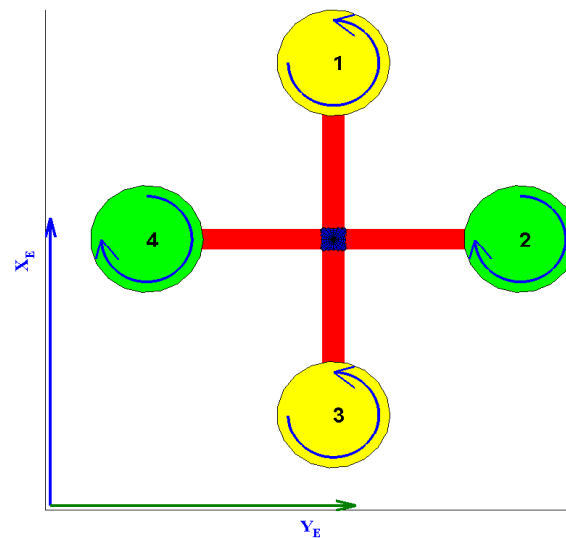


Figure 1. Diagram of a quadrotor, top view.

To enable autonomous operations of a quadrotor, the research community typically focuses on the following aspects of the quadrotor: kinematics and dynamics of the vehicle; trajectory generation (path planning); guidance, navigation and control.

The focus of this paper is to synthesize a nonlinear model predictive control law for tracking a trajectory to be followed by a quadrotor. The applications are typically way-point navigation and tracking certain agile maneuvers [7–10]. Since the quadrotor can only generate forces normal to the plane that contains the rotors, the translational and rotational dynamics are significantly coupled. This restricts the quadrotor from performing certain maneuvers [11] that are possible with a fixed wing aircraft and the more conventional helicopter. In what follows, some key works relevant to these topics are summarized, and the layout of the paper is outlined.

A significant amount of research has been performed in the area of trajectory generation and constrained control of unmanned vehicles that apply to quadrotors alike. The work in [12,13] discusses the generation of trajectories for robots, wherein a spline interpolation method is used to solve a minimum time optimization problem, while staying under torque and velocity constraints. The work in [14] also used a spline interpolation method to generate optimal-time trajectories and applied it to a micro-quadrotor. For reference trajectory generation, [7] develops the governing equations of motion and describes the trajectories as algebraic functions of a flat output: outputs that can express the states and inputs of the system in terms of its outputs and their derivatives [15]. This approach facilitates the automatic trajectory generation for the system. In [9], trajectories are generated by designing a sequence of controllers to drive the system to a desired goal state. The works in [1,16] discuss the design of safe, aggressive maneuvers and control for a back flip trajectory. The work in [17] constructed a dynamically feasible, desired speed profile for a given sequence of waypoints. The authors in [18,19] worked on trajectory generation for quadrotors by implementing dynamic constraints to an optimal control method and verified the existence of optimal trajectories. The work in [20] addresses the problem of quadrotor trajectory generation and tracking while carrying a suspended payload. They solved this by developing an optimal controller based on the dynamic programming technique: breaking down a problem into several sub-problems. It should be recognized that translational control of the quadcopter is very strongly coupled with the rigid body rotation especially for the longitudinal and lateral motion. The stability of such platforms has been studied, and some notable techniques that address the stability by separating the fast and slow dynamics can be found in [21,22].

Virtually every possible control technique, linear control, such as PID and linear quadratic methods, robust linear control and nonlinear control [23,24] techniques, such as adaptive control,

iterative learning control [25], neural networks [26], sliding mode, among others, have been tried and tested in simulations and on actual flying test platforms. Since the quadrotor can only produce bounded forces and that too only along one direction, the platform is input constrained. All of the earlier mentioned control techniques apply the input constraints post-facto, while model predictive control (MPC) provides a framework to impose input constraints as part of the control synthesis. Since the quadrotor is also constrained in operation in state-space, the platform essentially is state and input constrained. These state and input constraints along with goal states, as well as trajectory constraints (equality and inequality) can all be effectively integrated into the MPC framework. The work in [27] looked into the nonlinear model predictive tracking control (NMPTC) technique and applied it to generate trajectories to unmanned rotorcraft while staying under input and state constraints.

This paper focuses on the development of the nonlinear model predictive control (NMPC) formulation to derive an optimal controller subject to input, state and output constraints. This is accomplished by formulating the nonlinear system dynamics using a state-dependent coefficient (SDC) form, which allows the representation of the system in a pseudo-linear form. The performance of this controller is illustrated via candidate simulations for tracking aggressive reference trajectories in the presence of high frequency disturbances in the roll and pitch channels.

The rest of the paper is organized as follows. A brief overview of the governing equations of motion is first presented. This is followed by elaborating the MPC technique for a linearized quadrotor model (about the hover equilibrium) for the sake of completeness. The SDC form of the nonlinear quadrotor model is presented next, which highlights key differences in the presented model from those in other comparable works. Following this, the state-dependent Riccati equation (SDRE) approach to deriving a nonlinear sub-optimal controller is presented that uses the aforementioned SDC form. The nonlinear model predictive controller (NMPC) is developed using a discretized form (sampled-data form) of the SDC model. The boundedness of the trajectory tracking errors is shown for the NMPC, and simulation results are presented that compare the performance of linear and nonlinear MPC techniques, highlighting key areas where the nonlinear controller performs better. The simulation results correspond to representative maneuver scenarios. Finally, a brief summary of the work and conclusions thereof are presented.

2. Mathematical Model of the Quadrotor

Let $\{O_E, X_E, Y_E, Z_E\}$ denote the Earth-fixed inertial frame and $\{O_B, X_B, Y_B, Z_B\}$ the body-fixed frame, whose origin O_B is at the center of mass of the quadrotor. The inertial position of the quadrotor is defined by $\mathbf{p} = [x, y, z]^T$ and the attitude by the Euler angles: roll, pitch and yaw ($\Theta = [\phi, \theta, \psi]^T$). $\mathbf{R}_{BI} \in SO(3)$ is the direction cosine matrix representing the inertial to body frame transformation. $\mathbf{V}_B = [V_x, V_y, V_z]^T$ is the inertial velocity of the quadrotor expressed in the body frame components, and $\boldsymbol{\Omega} = [p, q, r]^T$ is the angular velocity expressed in the body frame components. Neglecting the aerodynamic and gyroscopic effects, the quadrotor model can be shown as [28]:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{R}_{BI}^T \mathbf{V}_B \\ \dot{\mathbf{V}}_B = -\boldsymbol{\Omega} \times \mathbf{V}_B + \mathbf{R}_{BI}(g\hat{\mathbf{e}}_3) + \frac{T}{m}\hat{\mathbf{e}}_3 \\ \dot{\Theta} = \mathbf{W}(\phi, \theta, \psi)\boldsymbol{\Omega} \\ \dot{\boldsymbol{\Omega}} = \mathbf{J}^{-1}(-\boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \boldsymbol{\tau}) \end{cases} \quad (1)$$

where $\hat{\mathbf{e}}_3 = [0 \ 0 \ 1]^T$, $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3]^T$, and:

$$\mathbf{R}_{BI}(\phi, \theta, \psi) = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \cos \psi \sin \phi - \sin \psi \cos \phi & \sin \theta \sin \psi \sin \phi + \cos \psi \cos \phi & \cos \theta \sin \phi \\ \sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi & \sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

$$\mathbf{W}(\phi, \theta, \psi) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} J_x & & \\ & J_y & \\ & & J_z \end{bmatrix} \quad (2)$$

The constants g , m , $\mathbf{J}$ in the equation denote the acceleration due to gravity, the mass and moment of inertia of the quadrotor, respectively. The body axes are assumed oriented along the principal axes without loss of any generality, and hence, the moment of inertia matrix is a diagonal matrix, $\mathbf{J} = \text{diag}(J_x, J_y, J_z)$. T and $\boldsymbol{\tau}$ represent the total thrust and torques about the body axes of the quadrotor.

In addition, the relations between the total thrust (T), torque ($\boldsymbol{\tau}$) components and the individual rotor thrusts (F_1, F_2, F_3, F_4) can be expressed as:

$$\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 0 & -L & 0 & L \\ L & 0 & -L & 0 \\ -c & c & -c & c \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \quad (3)$$

where L is the distance from the rotor to the center of gravity (CG) of the quadrotor and c is a constant that relates the rotor angular momentum to the rotor thrust (normal force). Combining Equations (1) and (3), a more comprehensive model can be written as:

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{V}}_{\mathbf{B}} \\ \dot{\boldsymbol{\Theta}} \\ \dot{\boldsymbol{\Omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{BI}^T \mathbf{V}_{\mathbf{B}} \\ -\boldsymbol{\Omega} \mathbf{V}_{\mathbf{B}} + \mathbf{R}_{BI}(g\hat{\mathbf{e}}_3) \\ \mathbf{W}(\phi, \theta, \psi)\boldsymbol{\Omega} \\ \mathbf{J}^{-1}(-\boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega}) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & -\frac{1}{m} & -\frac{1}{m} & -\frac{1}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -J_x^{-1}L & 0 & J_x^{-1}L \\ J_y^{-1}L & 0 & -J_y^{-1}L & 0 \\ -J_z^{-1}c & J_z^{-1}c & -J_z^{-1}c & J_z^{-1}c \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \quad (4)$$

Let $\mathbf{x} = [\mathbf{p}^T \mathbf{V}_{\mathbf{B}}^T \boldsymbol{\Theta}^T \boldsymbol{\Omega}^T]^T$ and $\mathbf{u} = [F_1 F_2 F_3 F_4]^T$. Equation (4) can be compactly written as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}_c \mathbf{u} \quad (5)$$

Given the nature of the quadrotor, the range of operation of the vehicle is restricted as shown below. At a pitch angle of $\theta = \pm \frac{\pi}{2}$, the matrix $\mathbf{W}(\phi, \theta, \psi)$ in Equation (2) becomes singular.

$$\begin{aligned} -\pi &\leq \phi \leq \pi \\ -\frac{\pi}{2} &< \theta < \frac{\pi}{2} \\ -\pi &< \psi < \pi \end{aligned} \quad (6)$$

Actuator dynamics for each rotor is assumed to be a first order system [29]:

$$\dot{F}_i = \lambda_F (F_i^c - F_i), \quad i = 1, 2, 3, 4. \quad (7)$$

where F_i^c denotes the commanded value for rotor lift (thrust) and λ_F denotes the first order actuator time constant (assumed the same for all of the rotors). Actuator dynamics is ignored for prediction, but is used for state propagation.

3. Linear Model Predictive Control of a Quadrotor

This section summarizes the model predictive control (MPC) algorithm for the sake of completeness. MPC, which can also be called receding horizon control (RHC), is a technique in which a mathematical model of a system is used to solve a finite, moving horizon, closed loop optimal control problem [30] by using the current states of the system [31]. MPC is able to take into account the physical and mechanical limitations of the plant during the design process [32] and predict a number of future outputs of the system (called the prediction horizon), in order to formulate an optimal controller effort to bring the system to a desired state given a reference trajectory.

This optimization problem is solved at each sampling interval for the specified prediction horizon, but only the first step of the solution to the optimization problem is applied to the system until the next sampling interval. This routine is repeated for all subsequent time intervals (see [33]). In the following section, the MPC technique as applied to a linear system (LMPC) is elaborated.

3.1. Plant Model and Prediction Horizon

Given the nonlinear system of the form:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t))\end{aligned}\quad (8)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ are the system states, $\mathbf{u}(t) \in \mathbb{R}^m$ are the system inputs and $\mathbf{y}(t) \in \mathbb{R}^p$ are the system outputs as functions of time t . In general $m < n$ and $m < p$. $\mathbf{f}$ and $\mathbf{h}$ are vector valued functions in $\mathcal{C}^2$. Note, the combined quadrotor governing equations of motion (Equation (4)) constitute such a nonlinear dynamic system where $n = 12$ and $m = 4$. This system can be linearized about a desired operating point, such as “hover”. The values of the states and the controls for such an operating point will be denoted as the ordered pair, $\{\mathbf{x}_T, \mathbf{u}_T\}$. The equivalent discretized linear system assuming a sampling interval T_s can be summarized as follows,

$$\begin{aligned}\Delta \mathbf{x}_{k+1} &= \mathbf{A} \Delta \mathbf{x}_k + \mathbf{B} \Delta \mathbf{u}_k \\ \Delta \mathbf{y}_k &= \mathbf{C} \Delta \mathbf{x}_k + \mathbf{D} \Delta \mathbf{u}_k\end{aligned}\quad (9)$$

where k is the current sample, $\Delta \mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_T$, $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_T$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the state matrix, $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the input matrix, $\mathbf{C} \in \mathbb{R}^{p \times n}$ is the output matrix and $\mathbf{D} \in \mathbb{R}^{p \times m}$ is the input feedforward matrix. The equilibrium control input in the case of “hover” is $\mathbf{u}_T = \left[\frac{mg}{4} \frac{mg}{4} \frac{mg}{4} \frac{mg}{4} \right]^T$.

The objective of the linear MPC is to drive the system towards a desired state. Based on the model of Equation (9) [34], the controller predicts the future states progression as a function of current states and future inputs:

$$\begin{aligned}\Delta \mathbf{x}_{k+i+1} &= \mathbf{A} \Delta \mathbf{x}_{k+i} + \mathbf{B} \Delta \mathbf{u}_{k+i} \\ \Delta \mathbf{y}_{k+i} &= \mathbf{C} \Delta \mathbf{x}_{k+i} + \mathbf{D} \Delta \mathbf{u}_{k+i}\end{aligned}\quad (10)$$

where $i = 1, 2, \dots, N$. Thus, the standard prediction equations can be written as,

$$\begin{aligned}\Delta \mathbf{X}_k &= \mathbf{F} \Delta \mathbf{x}_k + \mathbf{H} \Delta \mathbf{U}_k \\ \Delta \mathbf{Y}_k &= \bar{\mathbf{C}} \Delta \mathbf{X}_k + \bar{\mathbf{D}} \Delta \mathbf{U}_k\end{aligned}\quad (11)$$

where,

$$\Delta \mathbf{X}_k = \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{x}_{k+1} \\ \Delta \mathbf{x}_{k+2} \\ \vdots \\ \Delta \mathbf{x}_{k+N-1} \end{bmatrix}, \quad \Delta \mathbf{U}_k = \begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \mathbf{u}_{k+1} \\ \Delta \mathbf{u}_{k+2} \\ \vdots \\ \Delta \mathbf{u}_{k+N-1} \end{bmatrix}, \quad \Delta \mathbf{Y}_k = \begin{bmatrix} \Delta \mathbf{y}_k \\ \Delta \mathbf{y}_{k+1} \\ \Delta \mathbf{y}_{k+2} \\ \vdots \\ \Delta \mathbf{y}_{k+N-1} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^{N-1} \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}^{N-2}\mathbf{B} & \mathbf{A}^{N-3}\mathbf{B} & \cdots & \mathbf{B} & \mathbf{0} \end{bmatrix}, \quad \bar{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & & & & \\ & \mathbf{C} & & & \\ & & \mathbf{C} & & \\ & & & \ddots & \\ & & & & \mathbf{C} \end{bmatrix}, \quad \bar{\mathbf{D}} = \begin{bmatrix} \mathbf{D} & & & & \\ & \mathbf{D} & & & \\ & & \mathbf{D} & & \\ & & & \ddots & \\ & & & & \mathbf{D} \end{bmatrix}$$

Since $\mathbf{D} = \mathbf{0}$ in most applications, it is assumed for the rest of the analysis:

$$\begin{aligned} \Delta \mathbf{X}_k &= \mathbf{F} \Delta \mathbf{x}_k + \mathbf{H} \Delta \mathbf{U}_k \\ \Delta \mathbf{Y}_k &= \bar{\mathbf{C}} \Delta \mathbf{X}_k \end{aligned} \quad (12)$$

The terminal state and output are given by:

$$\Delta \mathbf{x}_{k+N} = \mathbf{A}^N \Delta \mathbf{x}_k + \bar{\mathbf{B}} \Delta \mathbf{U}_k, \quad \Delta \mathbf{y}_{k+N} = \mathbf{C} \Delta \mathbf{x}_{k+N}$$

where:

$$\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} \end{bmatrix}$$

3.2. Controller Design

The MPC algorithm requires the use of an objective function in its control formulation in order to calculate the optimal solution at each sampling interval. It must be chosen in a way such that the predicted outputs, derived from the prediction horizon N (Equation (11)), are driven to a desired state or track a desired trajectory $\mathbf{y}_k^r$, while at the same time, it should minimize the controller effort $\Delta \mathbf{u}_k$ required [35]. For the quadrotor scenario, the penalty function penalizes the norm of the difference between the current output states and the desired trajectory and the norm of actuator inputs. It is of the form:

$$J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k) = (\Delta \mathbf{Y}_k - \Delta \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N (\Delta \mathbf{Y}_k - \Delta \mathbf{Y}_k^r) + \Delta \mathbf{U}_k^T \bar{\mathbf{R}}_N \Delta \mathbf{U}_k + (\Delta \mathbf{y}_{k+N} - \Delta \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f (\Delta \mathbf{y}_{k+N} - \Delta \mathbf{y}_{k+N}^r) \quad (13)$$

It is important to note that usually, the reference trajectory $\Delta \mathbf{y}_{k+i}^r = \mathbf{y}_{k+i}^r - \mathbf{y}_T$, $i \in (1, \dots, N)$ is known in advance. Furthermore, $\Delta \mathbf{Y}_k^r$ is defined as:

$$\Delta \mathbf{Y}_k^r = \begin{bmatrix} \Delta \mathbf{y}_k^r \\ \Delta \mathbf{y}_{k+1}^r \\ \vdots \\ \Delta \mathbf{y}_{k+N-1}^r \end{bmatrix}$$

This implies that the controller is able to predict a series of adequate inputs that will drive the system towards the desired goal. In the quadratic form of (Equation (13)), the term $\bar{\mathbf{Q}}_N$ and $\bar{\mathbf{R}}_N$ are defined as:

$$\bar{\mathbf{Q}}_N = \begin{bmatrix} \mathbf{Q} & & & \\ & \mathbf{Q} & & \\ & & \ddots & \\ & & & \mathbf{Q} \end{bmatrix}, \quad \bar{\mathbf{R}}_N = \begin{bmatrix} \mathbf{R} & & & \\ & \mathbf{R} & & \\ & & \ddots & \\ & & & \mathbf{R} \end{bmatrix}$$

where $\mathbf{Q}, \mathbf{Q}_f \in \mathbb{R}^{p \times p}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ are all diagonal, and they satisfy $\mathbf{Q} \geq \mathbf{0}$, $\mathbf{Q}_f \geq \mathbf{0}$ and $\mathbf{R} > \mathbf{0}$. $\mathbf{Q}_f$ is chosen based on the solution to the discrete algebraic Riccati equation using $\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} + \mathbf{Q}$. $\mathbf{Q}_f$ is set equal to $\mathbf{P}$. With proper substitution, it follows that,

$$\begin{aligned} J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k) &= \Delta \mathbf{U}_k^T \left[\mathbf{H}^T \bar{\mathbf{C}}^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right] \Delta \mathbf{U}_k \\ &+ 2 \left((\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k - \Delta \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k - \Delta \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right) \Delta \mathbf{U}_k \\ &+ (\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k - \Delta \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N (\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k - \Delta \mathbf{Y}_k^r) \\ &+ (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k - \Delta \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k - \Delta \mathbf{y}_{k+N}^r) \end{aligned} \quad (14)$$

Quadratic programming: Since the cost function from Equation (14) is of quadratic form, a quadratic programming (QP) method can be used to solve the optimization problem [36]. The idea behind QP is to minimize the quadratic function $J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k)$ in Equation (14) by looking for a feasible search direction $\Delta \mathbf{U}_k$.

Input and state constraint handling: Special attention is given to the constraint handling capabilities of the MPC problem formulation now that the objective function has been specified (Equation (14)). In the case of the quadrotor, it is necessary to constrain both the total thrust force of each rotor and restrict the magnitude of the angles in order to stay within the limits allowed by the Euler angles' formulation discussed previously.

3.3. Input Constraints

It is necessary to constrain the maximum force each rotor is able to deliver in order to make the quadrotor perform in such a way that it resembles a more realistic physical model. The forces need to be within some lower bound $\mathbf{u}_{lb}$ and some upper bound $\mathbf{u}_{ub}$, $\mathbf{u}_{lb} \leq \mathbf{u}_{k+i} \leq \mathbf{u}_{ub}$ for $i = 0, 1, 2, \dots, N-1$. However, the MPC problem is solved to obtain the perturbation controls since the model employed is the linearized dynamics about the equilibrium point. Since, $\mathbf{u}_{k+i} = \mathbf{u}_T + \Delta \mathbf{u}_{k+i}$, it can be seen that $\mathbf{u}_{lb} \leq \mathbf{u}_T + \Delta \mathbf{u}_{k+i} \leq \mathbf{u}_{ub}$. Alternately, $\mathbf{u}_{lb} - \mathbf{u}_T \leq \Delta \mathbf{u}_{k+i} \leq \mathbf{u}_{ub} - \mathbf{u}_T$. These constraints can be expressed in matrix form as follows:

$$\begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} \Delta \mathbf{u}_{k+i} \leq \begin{bmatrix} (\mathbf{u}_{ub} - \mathbf{u}_T) \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \quad (15)$$

where $\mathbf{I}_{m \times m}$ is an $m \times m$ identity matrix. After proper arrangement, the input constraint can be represented as:

$$\mathbf{M}_U \Delta \mathbf{U}_k \leq \Delta \mathbf{U}_b \quad (16)$$

where:

$$\mathbf{M}_U = \begin{bmatrix} \begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} \\ \begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} \\ \ddots \\ \begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} \end{bmatrix}, \quad \Delta \mathbf{U}_b = \begin{bmatrix} \begin{bmatrix} (\mathbf{u}_{ub} - \mathbf{u}_T) \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \\ \begin{bmatrix} (\mathbf{u}_{ub} - \mathbf{u}_T) \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\mathbf{u}_{ub} - \mathbf{u}_T) \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \end{bmatrix} \quad (17)$$

3.4. Output and State Constraints

It is also of high importance to limit the angles so that no kinematic singularities are encountered due to the limitations in the model description. In the case of the quadrotor, the angles need to be within the bounds specified by Equation (6). If a particular output, such as the pitch angle, θ needs to be constrained, the corresponding output is defined as $\Delta z = \mathbf{C}_z \Delta \mathbf{x}_k$. The constraints are represented as $\mathbf{z}_{lb} \leq \mathbf{C}_z \mathbf{x}_{k+i} \leq \mathbf{z}_{ub}$ for $i = 0, 1, 2, \dots, N-1$, where $\mathbf{z}_{lb}$ and $\mathbf{z}_{ub}$ are the upper and lower bounds for the output variables. Since $\mathbf{x}_{k+i} = \mathbf{x}_T + \Delta \mathbf{x}_{k+i}$, $\mathbf{z}_{lb} \leq \mathbf{C}_z (\mathbf{x}_T + \Delta \mathbf{x}_{k+i}) \leq \mathbf{z}_{ub}$. Thus, $\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T \leq \mathbf{C}_z \Delta \mathbf{x}_{k+i} \leq \mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T$. It can be represented in matrix form as:

$$\begin{bmatrix} \mathbf{C}_z \\ -\mathbf{C}_z \end{bmatrix} \Delta \mathbf{x}_{k+i} \leq \begin{bmatrix} (\mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T) \\ -(\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T) \end{bmatrix} \quad (18)$$

With proper arrangement and substitution of $\Delta \mathbf{x}_k$ from (12), the constraints can be expressed in terms of $\Delta \mathbf{U}_k$ as:

$$\mathbf{C}_z (\mathbf{F} \Delta \mathbf{x}_k + \mathbf{H} \Delta \mathbf{U}_k) \leq \Delta \mathbf{Z}_b \quad (19)$$

where:

$$\mathbf{C}_z = \begin{bmatrix} \begin{bmatrix} \mathbf{C}_z \\ -\mathbf{C}_z \end{bmatrix} \\ \begin{bmatrix} \mathbf{C}_z \\ -\mathbf{C}_z \end{bmatrix} \\ \ddots \\ \begin{bmatrix} \mathbf{C}_z \\ -\mathbf{C}_z \end{bmatrix} \end{bmatrix}, \quad \Delta \mathbf{Z}_b = \begin{bmatrix} \begin{bmatrix} (\mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T) \\ -(\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T) \end{bmatrix} \\ \begin{bmatrix} (\mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T) \\ -(\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T) \\ -(\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T) \end{bmatrix} \end{bmatrix} \quad (20)$$

3.5. Combined Input and Output State Constraints

Both the input and output variable constraints can be integrated into one equation as shown below.

$$\begin{aligned} \mathbf{M}_U \Delta \mathbf{U}_k &\leq \Delta \mathbf{U}_b && \text{from Equation (16)} \\ \mathbf{C}_z (\mathbf{F} \Delta \mathbf{x}_k + \mathbf{H} \Delta \mathbf{U}_k) &\leq \Delta \mathbf{Z}_b && \text{from Equation (19)} \\ \text{or} \quad \mathbf{C}_z \mathbf{H} \Delta \mathbf{U}_k &\leq \Delta \mathbf{Z}_b - \mathbf{C}_z \mathbf{F} \Delta \mathbf{x}_k \\ \therefore \quad \mathbf{I} \Delta \mathbf{U}_k &\leq \mathbf{Y} \end{aligned} \quad (21)$$

where $\mathbf{Y}$ is a matrix containing both input and output variable constraints and:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{M}_U \\ \mathbf{C}_z \mathbf{H} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \Delta \mathbf{U}_b \\ \Delta \mathbf{Z}_b - \mathbf{C}_z \mathbf{F} \Delta \mathbf{x}_k \end{bmatrix}$$

3.6. Stability of Unconstrained Linear MPC

Without loss of generality, an “informal” proof of stability will be given only for the regulator case. For trajectory tracking problems, an error system can always be constructed with the new state being $\mathbf{e} = \mathbf{x} - \mathbf{x}^r$. With the proper feed-forward input, the new system would be of the same form as the regulator.

The cost function in Equation (14) can be simplified respectively as:

$$\begin{aligned} J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k) &= \Delta \mathbf{U}_k^T \left(\mathbf{H}^T \bar{\mathbf{C}}^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right) \Delta \mathbf{U}_k \\ &+ 2 \left((\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k)^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right) \Delta \mathbf{U}_k \\ &+ (\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k)^T \bar{\mathbf{Q}}_N (\bar{\mathbf{C}} \mathbf{F} \Delta \mathbf{x}_k) + (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k)^T \mathbf{Q}_f (\mathbf{C} \mathbf{A}^N \Delta \mathbf{x}_k) \end{aligned} \quad (22)$$

Assuming there are no constraints present on the inputs or states, the optimal control $\Delta \mathbf{U}_k^*$ can be determined by setting $\frac{\partial J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k)}{\partial \Delta \mathbf{U}_k} = \mathbf{0}$, i.e.,

$$\Delta \mathbf{U}_k^* = - \left(\mathbf{H}^T \bar{\mathbf{C}}^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right)^{-1} \left[(\bar{\mathbf{C}} \mathbf{H})^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{F} + (\mathbf{C} \bar{\mathbf{B}})^T \mathbf{Q}_f \mathbf{C} \mathbf{A}^N \right] \Delta \mathbf{x}_k \quad (23)$$

Additionally, the control input $\Delta \mathbf{u}_k^*$ can be extracted as follows.

$$\Delta \mathbf{u}_k^* = [\mathbf{I}_{m \times m} \quad \mathbf{0}_{m \times m} \quad \cdots \quad \mathbf{0}_{m \times m}] \Delta \mathbf{U}_k^* \quad (24)$$

Combining Equations (23) and (24), $\Delta \mathbf{u}_k$ can be simplified.

$$\Delta \mathbf{u}_k^* = -\mathbf{K}_k \Delta \mathbf{x}_k \quad (25)$$

where:

$$\mathbf{K}_k = [\mathbf{I}_{m \times m} \quad \mathbf{0}_{m \times m} \quad \cdots \quad \mathbf{0}_{m \times m}] \left[\mathbf{H}^T \bar{\mathbf{C}}^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{H} + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right]^{-1} \left[(\bar{\mathbf{C}} \mathbf{H})^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}} \mathbf{F} + (\mathbf{C} \bar{\mathbf{B}})^T \mathbf{Q}_f \mathbf{C} \mathbf{A}^N \right] \quad (26)$$

According to [37], if $\mathbf{Q}_f$ in Equation (22) satisfies the following inequality:

$$\mathbf{C}^T \mathbf{Q}_f \mathbf{C} \geq \mathbf{C}^T \mathbf{Q} \mathbf{C} + \mathbf{K}_k^T \mathbf{R} \mathbf{K}_k + (\mathbf{A} - \mathbf{B} \mathbf{K}_k)^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} (\mathbf{A} - \mathbf{B} \mathbf{K}_k) \quad (27)$$

for some $\mathbf{K}_k \in \mathcal{R}^{m \times n}$, then the system of Equation (9) driven by control $\Delta \mathbf{u}_k^*$ of Equation (24) is stable (note, the input feedforward matrix $\mathbf{D} = 0$).

The optimal cost $J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^*)$ satisfies the following monotonicity condition:

$$J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{1*}, N+1) \leq J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{2*}, N) \quad (28)$$

where N denotes the prediction horizon and $\Delta \mathbf{U}_k^{1*}, \Delta \mathbf{U}_k^{2*}$ represent the optimal control derived from minimizing their respective cost functions. Imagine a new control input $\Delta \mathbf{U}_k^1$ by using $\Delta \mathbf{U}_k^{2*}$ up to time $k + N - 1$ and $\Delta \mathbf{u}_{k+N} = -\mathbf{K}_k \Delta \mathbf{x}_{k+N}$; the cost for this control would be:

$$J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^1, N+1) = \sum_{i=k}^{k+N-1} (\Delta \mathbf{x}_i^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_i + \Delta \mathbf{u}_i^T \mathbf{R} \Delta \mathbf{u}_i) + \Delta \mathbf{x}_{k+N}^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_{k+N} \\ + \Delta \mathbf{x}_{k+N}^T \mathbf{K}_k^T \mathbf{R} \mathbf{K}_k \Delta \mathbf{x}_{k+N} + \Delta \mathbf{x}_{k+N}^T (\mathbf{A} - \mathbf{B} \mathbf{K}_k)^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} (\mathbf{A} - \mathbf{B} \mathbf{K}_k) \Delta \mathbf{x}_{k+N} \quad (29)$$

Since $J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{1*}, N+1)$ is optimal and denoting $\Delta J = J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{1*}, N+1) - J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{2*}, N)$, the following applies:

$$\Delta J \leq J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^1, N+1) - J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^{2*}, N) \\ \leq \Delta \mathbf{x}_{k+N}^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_{k+N} + \Delta \mathbf{x}_{k+N}^T \mathbf{K}_k^T \mathbf{R} \mathbf{K}_k \Delta \mathbf{x}_{k+N} \\ + \Delta \mathbf{x}_{k+N}^T (\mathbf{A} - \mathbf{B} \mathbf{K}_k)^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} (\mathbf{A} - \mathbf{B} \mathbf{K}_k) \Delta \mathbf{x}_{k+N} - \Delta \mathbf{x}_{k+N}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \Delta \mathbf{x}_{k+N} \\ \leq \Delta \mathbf{x}_{k+N}^T [\mathbf{C}^T \mathbf{Q} \mathbf{C} + \mathbf{K}_k^T \mathbf{R} \mathbf{K}_k + (\mathbf{A} - \mathbf{B} \mathbf{K}_k)^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} (\mathbf{A} - \mathbf{B} \mathbf{K}_k) - \mathbf{C}^T \mathbf{Q}_f \mathbf{C}] \Delta \mathbf{x}_{k+N} \\ \leq 0 \quad (30)$$

thus proving the cost monotonicity condition. From the cost monotonicity condition, the following non-increasing sequence can be obtained:

$$J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^*, N) = \Delta \mathbf{x}_k^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_k + \Delta \mathbf{u}_k^T \mathbf{R} \Delta \mathbf{u}_k + J(\Delta \mathbf{x}_{k+1}, \Delta \mathbf{U}_{k+1}^*, N-1) \\ \geq \Delta \mathbf{x}_k^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_k + \Delta \mathbf{u}_k^T \mathbf{R} \Delta \mathbf{u}_k + J(\Delta \mathbf{x}_{k+1}, \Delta \mathbf{U}_{k+1}^*, N) \\ \geq J(\Delta \mathbf{x}_{k+1}, \Delta \mathbf{U}_{k+1}^*, N) \quad (31)$$

Since a non-increasing sequence bounded from below converges to a constant, $J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^*, N) \geq 0$, and hence, $J(\Delta \mathbf{x}_k, \Delta \mathbf{U}_k^*, N) \rightarrow$ a nonnegative constant as $k \rightarrow \infty$. Thus,

$$\sum_{k=i}^{i+j} (\Delta \mathbf{x}_k^T \mathbf{C}^T \mathbf{Q} \mathbf{C} \Delta \mathbf{x}_k + \Delta \mathbf{u}_k^T \mathbf{R} \Delta \mathbf{u}_k) \rightarrow 0, \quad j = 0, 1, 2, \dots \quad (32)$$

which leads to the following equation:

$$\sum_{k=i}^{i+j} \Delta \mathbf{x}_k^T [(\mathbf{A} - \mathbf{B} \mathbf{K}_k)^j]^T (\mathbf{C}^T \mathbf{Q} \mathbf{C} + \mathbf{K}_k^T \mathbf{R} \mathbf{K}_k) [(\mathbf{A} - \mathbf{B} \mathbf{K}_k)^j] \Delta \mathbf{x}_k \rightarrow 0, \quad j = 0, 1, 2, \dots \quad (33)$$

It is proven in [38] that if $(\mathbf{A}, \mathbf{C})$ is observable, then:

$$\left((\mathbf{A} - \mathbf{B} \mathbf{K}_k), \begin{bmatrix} \sqrt{\mathbf{Q} \mathbf{C}} \\ \sqrt{\mathbf{R} \mathbf{K}_k} \end{bmatrix} \right)$$

is observable for any $\mathbf{K}_k$. Thus, the only solution that could guarantee this is the trivial solution $\Delta \mathbf{x}_k = \mathbf{0}$. Hence, system Equation (9), driven by control $\Delta \mathbf{u}_k^*$ of Equation (24) by minimizing the quadratic cost Equation (22), is stable. Back to our control law design, if a $\mathbf{Q}_f$ is carefully selected such that $\mathbf{C}^T \mathbf{Q}_f \mathbf{C}$ is the solution of the Discrete Algebraic Riccati Equation (DARE), thus automatically satisfying the Equation (27), the optimal control given by Equation (25) asymptotically stabilizes system Equation (9). Since full state feedback is assumed for this problem ($\mathbf{C} = \mathbf{I}$), this can be easily accomplished.

3.7. Stability of Constrained Linear MPC

Again without loss of generality, only stability for the regulator case is considered here. To guarantee the same monotonicity condition as Equation (28), it must be guaranteed that the system is feasible. The following linear matrix inequalities (LMI) are constructed for this purpose. There must exist a set of $(\mathbf{Q}_f, \mathbf{Q}, \mathbf{R}, \mathbf{K}_k, \Delta \mathbf{U}_k)$ that satisfies the following conditions:

$$\gamma < \infty \quad (34)$$

$$\begin{bmatrix} \gamma - 2 \left[(\bar{\mathbf{C}}\mathbf{F}\Delta\mathbf{x}_k)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}\mathbf{H} + (\mathbf{C}\mathbf{A}^N \Delta\mathbf{x}_k)^T \mathbf{Q}_f \mathbf{C}\bar{\mathbf{B}} \right] \Delta\mathbf{U}_k & \Delta\mathbf{U}_k^T \\ \Delta\mathbf{U}_k & \left(\mathbf{H}^T \bar{\mathbf{C}}^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}\mathbf{H} + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}^T \mathbf{C}^T \mathbf{Q}_f \mathbf{C} \bar{\mathbf{B}} \right)^{-1} \end{bmatrix} \geq \mathbf{0} \quad (35)$$

$$\mathbf{\Gamma} \Delta\mathbf{U}_k \leq \mathbf{Y} \quad (36)$$

$$\begin{bmatrix} (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} & (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} (\mathbf{A} - \mathbf{B}\mathbf{K}_k)^T & (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} \sqrt{\mathbf{C}^T \mathbf{Q} \mathbf{C}} & (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} \mathbf{K}_k^T \sqrt{\mathbf{R}} \\ (\mathbf{A} - \mathbf{B}\mathbf{K}_k) (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} & (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \\ \sqrt{\mathbf{C}^T \mathbf{Q} \mathbf{C}} (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} & \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} & \mathbf{0}_{n \times n} \\ \sqrt{\mathbf{R}} \mathbf{K}_k (\mathbf{C}^T \mathbf{Q}_f \mathbf{C})^{-1} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} \end{bmatrix} \geq \mathbf{0} \quad (37)$$

$$\begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} \mathbf{K}_k (\mathbf{A}^N \Delta\mathbf{x}_k + \bar{\mathbf{B}} \Delta\mathbf{U}_k) \leq \begin{bmatrix} (\mathbf{u}_{ub} - \mathbf{u}_T) \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \quad (38)$$

$$\begin{bmatrix} \mathbf{C}_z \\ -\mathbf{C}_z \end{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{K}_k) (\mathbf{A}^N \Delta\mathbf{x}_k + \bar{\mathbf{B}} \Delta\mathbf{U}_k) \leq \begin{bmatrix} (\mathbf{z}_{ub} - \mathbf{C}_z \mathbf{x}_T) \\ -(\mathbf{z}_{lb} - \mathbf{C}_z \mathbf{x}_T) \end{bmatrix} \quad (39)$$

It should be noted here that $\mathbf{C}^T \mathbf{Q}_f \mathbf{C}$ needs to be full rank. Since the constraints contain constant terms, this condition only needs to be checked just once at the beginning. After obtaining a suitable $(\mathbf{Q}_f, \mathbf{Q}, \mathbf{R})$ from the above LMI, $\Delta\mathbf{U}_k^*$ is obtained by solving the following optimization problem:

$$\Delta\mathbf{U}_k^* = \arg \min_{\Delta\mathbf{U}_k} \gamma$$

subject to Equations (34)–(39).

The solution to the above-mentioned optimization problem yields $\Delta\mathbf{u}_k^*$,

$$\Delta\mathbf{u}_k^* = [\mathbf{I}_{m \times m} \ \mathbf{0}_{m \times m} \ \cdots \ \mathbf{0}_{m \times m}] \Delta\mathbf{U}_k^* \quad (40)$$

The same proof of asymptotic stability from the unconstrained case can then be used here once the feasible control action is determined. Thus, system Equation (9) driven by control Equation (40) subject to constraints Equation (21) is stable. Note, the stability of such platforms with coupled fast and slow dynamics has been studied extensively by other researchers, and some notable techniques, such as singular perturbation control can be found in [21,22]. In the present context, the translational control in the horizontal plane (forward and lateral directions) can only be affected by the pitch and roll angles. Thus, it is imperative for the quadrotor platform to have a very robust pitch and roll control. In our present work, we do not explicitly address this time scale decomposition, since the closed loop system is guaranteed stable with an appropriate choice of the weighting matrices. Robustness could be improved by further turning these weighting matrices.

4. Nonlinear Model Predictive Control Formulation

In this section, the procedure to develop the nonlinear model predictive controller (NMPC) is elaborated. The essential idea is to utilize the state-dependent coefficient (SDC) factorization [39] of the nonlinear dynamics. A state space representation of the quadrotor is obtained, where each of its system matrices are now expressed as functions of the current state [40]. Consider a dynamic system as

in Equation (8). The NMPC formulation involves transforming the nonlinear system into the following state space form,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_c(\mathbf{x})\mathbf{x} + \mathbf{B}_c(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{C}_c(\mathbf{x})\mathbf{x} + \mathbf{D}_c(\mathbf{x})\mathbf{u}\end{aligned}\quad (41)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector, $\mathbf{y} \in \mathbb{R}^p$ is the output vector and $\mathbf{A}_c(\mathbf{x}) \in \mathbb{R}^{n \times n}$, $\mathbf{B}_c(\mathbf{x}) \in \mathbb{R}^{n \times m}$, $\mathbf{C}_c(\mathbf{x}) \in \mathbb{R}^{p \times n}$ and $\mathbf{D}_c(\mathbf{x}) \in \mathbb{R}^{p \times m}$ are the pseudo-linear system matrices in the SDC form, as well as the pairs $(\mathbf{A}_c(\mathbf{x}), \mathbf{B}_c(\mathbf{x}))$ and $(\mathbf{A}_c(\mathbf{x}), \mathbf{C}_c(\mathbf{x}))$ are stabilizable and detectable $\forall \mathbf{x} \in \mathbb{R}^n$, respectively. It is important to note that this representation of $\mathbf{A}_c(\mathbf{x})$, $\mathbf{B}_c(\mathbf{x})$, $\mathbf{C}_c(\mathbf{x})$ and $\mathbf{D}_c(\mathbf{x})$ is not unique in general, except for a scalar system. Different state-dependent coefficient matrices can be obtained from the equations of motion, and a solution to the optimization problem may or may not exist. However, a particular factorization (shown later) is derived here. The discrete-time equivalent of Equation (41) is obtained by using a zero-order-hold (ZOH) with a specified sample time. Let the discrete-time equivalent of the system be of the following form:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}(\mathbf{x}_k)\mathbf{x}_k + \mathbf{B}(\mathbf{x}_k)\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}(\mathbf{x}_k)\mathbf{x}_k\end{aligned}\quad (42)$$

where $\mathbf{A}(\mathbf{x}_k)$ and $\mathbf{B}(\mathbf{x}_k)$ are discrete approximations of the continuous $\mathbf{A}_c(\mathbf{x})$ and $\mathbf{B}_c(\mathbf{x})$, respectively.

From the description of the quadrotor, it can be seen that $\mathbf{D}(\mathbf{x}_k) = \mathbf{0}$ and $\mathbf{B}(\mathbf{x}_k) = \mathbf{B}$ (a constant matrix). Since Equation (42) is of the pseudo-linear form, the system matrices can be considered to be constants for each sampling interval, i.e., $[t_k, t_{k+1})$ with $t_{k+1} - t_k = \Delta t$. It should be noted that in Equation (41), the first term $\mathbf{A}_c(\mathbf{x})\mathbf{x}$ vanishes if $\mathbf{x} = \mathbf{0}$. For the quadrotor case (Equation (5)), $f(\mathbf{x})$ does not completely vanish when $\mathbf{x} = \mathbf{0}$. Thus, a slight change is needed in Equation (41) to account for this. The quadrotor system will be transformed into the form:

$$\dot{\mathbf{x}} = \mathbf{A}_c(\mathbf{x})\mathbf{x} + \mathbf{B}_c(\mathbf{x})(\mathbf{u}_0 + \delta u) \quad (43)$$

where $\mathbf{u}_0$ represents the constant force term that is required to balance the weight of the quadrotor. It is assumed that the quadrotor starts from an initial equilibrium state. The new control design then focuses on the synthesis of δu . Clearly, when $\delta u = \mathbf{0}$, for the “hover” equilibrium case, $\dot{\mathbf{x}} = \mathbf{0}$, since $\mathbf{A}_c(\mathbf{x})\mathbf{x} + \mathbf{B}_c(\mathbf{x})\mathbf{u}_0 = \mathbf{0}$. One possible way to factorize the equations of motion into the SDC form is presented in Equation (44).

$$\mathbf{A}_c(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{BI}^T & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{A}_{24} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{W} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{44} \end{bmatrix} \quad (44)$$

where:

$$\begin{aligned}\mathbf{A}_{22} &= \begin{bmatrix} 0 & \frac{r}{2} & -\frac{q}{2} \\ -\frac{r}{2} & 0 & \frac{p}{2} \\ \frac{q}{2} & -\frac{p}{2} & 0 \end{bmatrix}, \quad \mathbf{A}_{23} = \begin{bmatrix} 0 & -g \frac{\sin \theta}{\theta} & 0 \\ g \frac{\cos \theta \sin \phi}{\phi} & 0 & 0 \\ g \frac{(\cos \theta + 1)(\cos \phi - 1)}{2\phi} & g \frac{(\cos \phi + 1)(\cos \theta - 1)}{2\theta} & 0 \end{bmatrix} \\ \mathbf{A}_{24} &= \begin{bmatrix} 0 & -\frac{w}{2} & \frac{v}{2} \\ \frac{w}{2} & 0 & -\frac{u}{2} \\ -\frac{v}{2} & \frac{u}{2} & 0 \end{bmatrix}, \quad \mathbf{A}_{44} = \begin{bmatrix} 0 & \frac{(J_y - J_z)r}{2J_x} & \frac{(J_y - J_z)q}{2J_x} \\ \frac{(J_z - J_x)r}{2J_y} & 0 & \frac{(J_z - J_x)p}{2J_y} \\ \frac{(J_x - J_y)q}{2J_z} & \frac{(J_x - J_y)p}{2J_z} & 0 \end{bmatrix}\end{aligned}$$

and $\mathbf{B}_c$ is the same as in Equation (4). One issue that arises from this SDC form is that the term like $\frac{\sin \theta}{\theta}$ does not exist when $\theta = 0$. To prevent this from happening, the first three terms of Taylor series expansions of $\sin \theta$, $\sin \phi$, $(\cos \theta - 1)$ and $(\cos \phi - 1)$ are used here to provide a close approximation (in reality, the approximations only differ from their true values 4% at most in the range of $(-\frac{\pi}{2}, \frac{\pi}{2})$). By doing this, $\mathbf{A}_{23}$ can be expressed as shown in Equation (45).

$$\mathbf{A}_{23} = \begin{bmatrix} 0 & -g(1 - \frac{\theta^2}{3!} + \frac{\theta^4}{5!}) & 0 \\ g \cos \theta (1 - \frac{\phi^2}{3!} + \frac{\phi^4}{5!}) & 0 & 0 \\ g \frac{\cos \theta + 1}{2} (-\frac{\phi}{2} + \frac{\phi^3}{4!} - \frac{\phi^5}{6!}) & g \frac{\cos \phi + 1}{2} (-\frac{\theta}{2} + \frac{\theta^3}{4!} - \frac{\theta^5}{6!}) & 0 \end{bmatrix} \quad (45)$$

Considering the specific aspects of the quadrotor as detailed above, the discretized system for the derivation of the NMPC is summarized below:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}(\mathbf{x}_k) \mathbf{x}_k + \mathbf{B}(\mathbf{u}_T + \delta \mathbf{u}_k) \\ \mathbf{y}_k &= \mathbf{C}(\mathbf{x}_k) \mathbf{x}_k \end{aligned} \quad (46)$$

where the fact that for this system, $\mathbf{B}(\mathbf{x}_k) = \mathbf{B}$ (a constant matrix) is used. Following the same approach as outlined for the linear MPC, we arrive at the N step state prediction equations (similar to Equation (12)):

$$\begin{aligned} \mathbf{X}_k &= \mathbf{F}(\mathbf{x}_k) \mathbf{x}_k + \mathbf{H}(\mathbf{x}_k) (\mathbf{U}_T + \Delta \mathbf{U}_k) \\ \mathbf{Y}_k &= \bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{X}_k \end{aligned} \quad (47)$$

where,

$$\begin{aligned} \mathbf{X}_k &= \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \\ \mathbf{x}_{k+2} \\ \vdots \\ \mathbf{x}_{k+N-1} \end{bmatrix}, \quad \mathbf{U}_T = \begin{bmatrix} \mathbf{u}_T \\ \mathbf{u}_T \\ \mathbf{u}_T \\ \vdots \\ \mathbf{u}_T \end{bmatrix}, \quad \Delta \mathbf{U}_k = \begin{bmatrix} \delta \mathbf{u}_k \\ \delta \mathbf{u}_{k+1} \\ \delta \mathbf{u}_{k+2} \\ \vdots \\ \delta \mathbf{u}_{k+N-1} \end{bmatrix}, \quad \mathbf{Y}_k = \begin{bmatrix} \mathbf{y}_k \\ \mathbf{y}_{k+1} \\ \mathbf{y}_{k+2} \\ \vdots \\ \mathbf{y}_{k+N-1} \end{bmatrix} \\ \mathbf{F}(\mathbf{x}_k) &= \begin{bmatrix} \mathbf{I} \\ \mathbf{A}(\mathbf{x}_k) \\ \mathbf{A}(\mathbf{x}_k)^2 \\ \vdots \\ \mathbf{A}(\mathbf{x}_k)^{N-1} \end{bmatrix}, \quad \mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{B} & \mathbf{0} \\ \mathbf{A}(\mathbf{x}_k)\mathbf{B} & \mathbf{B} & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{A}(\mathbf{x}_k)^{N-2}\mathbf{B} & \mathbf{A}(\mathbf{x}_k)^{N-3}\mathbf{B} & \dots & \mathbf{B} & \mathbf{0} \end{bmatrix} \\ \bar{\mathbf{C}}(\mathbf{x}_k) &= \begin{bmatrix} \mathbf{C}(\mathbf{x}_k) & & & & \\ & \mathbf{C}(\mathbf{x}_k) & & & \\ & & \mathbf{C}(\mathbf{x}_k) & & \\ & & & \ddots & \\ & & & & \mathbf{C}(\mathbf{x}_k) \end{bmatrix} \end{aligned}$$

Similarly, the terminal state and output is given by:

$$\mathbf{x}_{k+N} = \mathbf{A}(\mathbf{x}_k)^N \mathbf{x}_k + \bar{\mathbf{B}}(\mathbf{x}_k) (\mathbf{U}_T + \Delta \mathbf{U}_k), \quad \mathbf{y}_{k+N} = \mathbf{C}(\mathbf{x}_k) \mathbf{x}_{k+N}$$

where:

$$\bar{\mathbf{B}}(\mathbf{x}_k) = \begin{bmatrix} \mathbf{A}(\mathbf{x}_k)^{N-1} \mathbf{B} & \mathbf{A}(\mathbf{x}_k)^{N-2} \mathbf{B} & \dots & \mathbf{A}(\mathbf{x}_k) \mathbf{B} & \mathbf{B} \end{bmatrix}$$

The nonlinear MPC uses the same algorithm as the linear MPC to calculate the solution to the minimization of the cost function outlined in Equation (13). The main difference between the linear and nonlinear version of the MPC is that now, the objective function depends on the current states of the system and needs to be calculated at the start of each sample interval. The objective function Equation (14) then becomes:

$$\begin{aligned}
 J(\mathbf{x}_k, \mathbf{U}_k) = & \Delta \mathbf{U}_k^T \left(\mathbf{H}(\mathbf{x}_k)^T \bar{\mathbf{C}}(\mathbf{x}_k)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}(\mathbf{x}_k)^T \mathbf{C}(\mathbf{x}_k)^T \mathbf{Q}_f \mathbf{C}(\mathbf{x}_k) \bar{\mathbf{B}}(\mathbf{x}_k) \right) \Delta \mathbf{U}_k \\
 & + 2 \left[(\bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{F}(\mathbf{x}_k) \mathbf{x}_k - \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) + (\mathbf{C}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k)^N \mathbf{x}_k - \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f \mathbf{C}(\mathbf{x}_k) \bar{\mathbf{B}}(\mathbf{x}_k) \right] \Delta \mathbf{U}_k \\
 & + \mathbf{U}_T^T \left(\mathbf{H}(\mathbf{x}_k)^T \bar{\mathbf{C}}(\mathbf{x}_k)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) + \bar{\mathbf{R}}_N + \bar{\mathbf{B}}(\mathbf{x}_k)^T \mathbf{C}(\mathbf{x}_k)^T \mathbf{Q}_f \mathbf{C}(\mathbf{x}_k) \bar{\mathbf{B}}(\mathbf{x}_k) \right) \mathbf{U}_T \\
 & + 2 \left[(\bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{F}(\mathbf{x}_k) \mathbf{x}_k - \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N \bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) + (\mathbf{C}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k)^N \mathbf{x}_k - \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f \mathbf{C}(\mathbf{x}_k) \bar{\mathbf{B}}(\mathbf{x}_k) \right] \mathbf{U}_T \\
 & + (\bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{F}(\mathbf{x}_k) \mathbf{x}_k - \mathbf{Y}_k^r)^T \bar{\mathbf{Q}}_N (\bar{\mathbf{C}}(\mathbf{x}_k) \mathbf{F}(\mathbf{x}_k) \mathbf{x}_k - \mathbf{Y}_k^r) \\
 & + (\mathbf{C}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k)^N \mathbf{x}_k - \mathbf{y}_{k+N}^r)^T \mathbf{Q}_f (\mathbf{C}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k)^N \mathbf{x}_k - \mathbf{y}_{k+N}^r)
 \end{aligned} \quad (48)$$

As with the linear case, $\mathbf{Q}_f$ is chosen based on the solution to the discrete algebraic Riccati equation using:

$$\mathbf{P}(\mathbf{x}_k) = \mathbf{A}(\mathbf{x}_k)^T \mathbf{P}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k) - \mathbf{A}(\mathbf{x}_k)^T \mathbf{P}(\mathbf{x}_k) \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P}(\mathbf{x}_k) \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}(\mathbf{x}_k) \mathbf{A}(\mathbf{x}_k) + \mathbf{Q} \quad (49)$$

and $\mathbf{Q}_f$ is set equal to $\mathbf{P}(\mathbf{x}_k)$. This calculation is performed at each sample instant as opposed to just once, which was the case in the linear control law derivation.

The cost function is seen to be quasi-quadratic, and the regular quadratic programming method is used to solve for the control (Equation (48)). Similar to Equation (21), the input and output variable constraints are integrated into one equation of the form:

$$\mathbf{\Gamma}(x_k) \Delta \mathbf{U}_k \leq \mathbf{Y}(x_k) \quad (50)$$

where $\mathbf{Y}(x_k)$ is a matrix containing both input and output variable constraints and:

$$\mathbf{\Gamma}(x_k) = \begin{bmatrix} \mathbf{M}_U \\ \mathbf{C}_z(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) \end{bmatrix}, \quad \mathbf{Y}(x_k) = \begin{bmatrix} \Delta \mathbf{U}_b \\ \mathbf{Z}_b - \mathbf{C}_z(\mathbf{x}_k) (\mathbf{F}(\mathbf{x}_k) \mathbf{x}_k + \mathbf{H}(\mathbf{x}_k) \mathbf{U}_T) \end{bmatrix}$$

where,

$$\Delta \mathbf{U}_b = \begin{bmatrix} \begin{bmatrix} \mathbf{u}_{ub} - \mathbf{u}_T \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \\ \begin{bmatrix} \mathbf{u}_{ub} - \mathbf{u}_T \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{u}_{ub} - \mathbf{u}_T \\ -(\mathbf{u}_{lb} - \mathbf{u}_T) \end{bmatrix} \end{bmatrix}, \quad \mathbf{Z}_b = \begin{bmatrix} \begin{bmatrix} \mathbf{z}_{ub} \\ -\mathbf{z}_{lb} \end{bmatrix} \\ \begin{bmatrix} \mathbf{z}_{ub} \\ -\mathbf{z}_{lb} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{z}_{ub} \\ -\mathbf{z}_{lb} \end{bmatrix} \end{bmatrix}$$

Note, the inequality in Equation (50) needs to be checked at the start of every new prediction. Alternately, the feasibility of the solution to the LMIs for NMPC needs to be evaluated at every new prediction.

5. Stability of Nonlinear Model Predictive Control Solution

In this section, the stability of the closed loop system with the NMPC solution from minimizing the cost function in Equation (48) subject to the input and state constraints in Equation (50), when applied to the dynamics in Equation (1), is shown. It is shown that the control scheme when applied to the quadrotor model results in bounded tracking errors, while guaranteeing internal stability. The bounded stability is shown for a general class of nonlinear systems representative of the application under consideration. Consider the nonlinear system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) \mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (51)$$

with the definition of the state and control variables to be the same as what was discussed in earlier sections. The mappings $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ are sufficiently smooth. A pseudo-linear description of Equation (51) is the pair $(\mathbf{A}_c(\mathbf{x}), \mathbf{B}_c(\mathbf{x}))$, such that,

$$\dot{\mathbf{x}} = \mathbf{A}_c(\mathbf{x})\mathbf{x} + \mathbf{B}_c(\mathbf{x})\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (52)$$

Since the control implementation essentially renders the closed loop system to be a sampled-data system, the stability problem is addressed by first considering the unconstrained sampled-data NMPC for the SDC system in Equation (52). The constrained sampled-data NMPC is considered later.

5.1. Stability of Unconstrained Sampled-Data NMPC Based on the SDC System

Assumptions:

- $(\mathbf{A}_c(\mathbf{x}), \mathbf{B}_c(\mathbf{x}))$ is point-wise controllable. Thus, $\forall \mathbf{x} \in \mathbb{R}^n$, i.e., $\exists \mathbf{K}_c(\mathbf{x}) \in \mathbb{R}^{m \times n}$, such that $(\mathbf{A}_c(\mathbf{x}) - \mathbf{B}_c(\mathbf{x})\mathbf{K}_c(\mathbf{x}))$ is point-wise Hurwitz.
- $\mathbf{K}_c(\mathbf{x})$ is obtained as a solution to a state-dependent Riccati equation using the system matrices in the SDC form. The control law is thus expressed as,

$$\mathbf{u}_c = -\mathbf{K}_c(\mathbf{x})(\mathbf{x} - \mathbf{x}^r) \quad (53)$$

- $\mathbf{x}^r(t)$ is an admissible reference state trajectory, i.e., $\mathbf{x}^r(t)$ satisfies the governing equations of motion, as well as the state constraints.
- The reference control inputs $\mathbf{u}^r(t)$ obtained from Equation (51) together with $\mathbf{x}^r(t)$ and $\dot{\mathbf{x}}^r(t)$ satisfy the control constraints discussed previously.
- The solution is feasible for the given predictive horizon.

In this section, following a similar approach reported in [41], it will be first shown that for an appropriate sample time Δt , the ZOH control, computed as:

$$\mathbf{u}_c(k\Delta t) = -\mathbf{K}_c(\mathbf{x}(k\Delta t))(\mathbf{x}(k\Delta t) - \mathbf{x}^r(k\Delta t))$$

when applied to the nonlinear system in Equation (51) results in bounded trajectory tracking errors, i.e., $\|\mathbf{x}(k\Delta t) - \mathbf{x}^r(k\Delta t)\| < \varepsilon$, $\varepsilon > 0$ for $k > k_N$, where $k_N \in \mathbb{Z}^+$. For the rest of the discussion, $\mathbf{x}(k\Delta t)$ and $\mathbf{u}_c(k\Delta t)$ will be simply written as $\mathbf{x}_k$ and $\mathbf{u}_k$.

For $t \in [k\Delta t, (k+1)\Delta t]$, the frozen in time SDC representation for the system is considered; thus, the states evolve as,

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}_c(\tilde{\mathbf{x}}_k)\tilde{\mathbf{x}} + \mathbf{B}_c(\tilde{\mathbf{x}}_k)\mathbf{u}_k, \quad \tilde{\mathbf{x}}(k\Delta t) = \mathbf{x}_k, \quad t \in [k\Delta t, (k+1)\Delta t] \quad (54)$$

At the next sampling interval $[(k+1)\Delta t, (k+2)\Delta t]$, $\tilde{\mathbf{x}}((k+1)\Delta t)$ is replaced with $\mathbf{x}_{k+1}$ measured from the original system Equation (52), and the same process starts over again. To avoid potential ambiguities, the solution of differential Equation (54) at the end of the interval, namely the value of $\tilde{\mathbf{x}}$ as $t \rightarrow (k+1)\Delta t$, is denoted as $\tilde{\mathbf{x}}'_{k+1}$. Note, the control input $\mathbf{u}_c$ is constant during the interval $[k\Delta t, (k+1)\Delta t]$. Under the assumptions stated earlier, the control for Equation (54) will stabilize the original system Equation (52) provided the control law given by Equation (53) will achieve uniformly globally asymptotic stability (UGAS) for System (51). It should be noted here that this algorithm is merely a sampled-data implementation of Equation (53). The work in [41] proved the stability of sampled-data control based on SDC provided that $\mathbf{P}_k$ obtained from Equation (49) converges to a constant matrix; in other words, $\lim_{k \rightarrow \infty} \mathbf{P}_k$ exists. However, the convergence of $\mathbf{P}_k$ is hard, if not impossible to guarantee. Here, an alternative proof with a different, albeit restrictive assumption, is given. Without loss of generality, only the stability of the system for the regulator case, meaning $\mathbf{x}^r = \mathbf{0}$ at all times, is proven. Firstly, $\mathbf{x}_k \neq \mathbf{0}$ since $\mathbf{x}_k = \mathbf{0}$ dictates $\mathbf{u}_k = \mathbf{0}$, which makes it trivial. With a constant input $\mathbf{u}_k$ during time interval $t \in [k\Delta t, (k+1)\Delta t]$, $\mathbf{x}_{k+1}$ is obtained from the following:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} [\mathbf{f}(\mathbf{x}(\tau)) + \mathbf{g}(\mathbf{x}(\tau))\mathbf{u}_k] d\tau \quad (55)$$

and obtain $\tilde{\mathbf{x}}'_{k+1}$ from:

$$\tilde{\mathbf{x}}'_{k+1} = e^{\mathbf{A}_c(\mathbf{x}_k)\Delta t}\mathbf{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} \left(e^{\mathbf{A}_c(\mathbf{x}_k)(s-\tau)} \mathbf{B}_c(\mathbf{x}_k)\mathbf{u}_k \right) d\tau \quad (56)$$

The control input $\mathbf{u}_k$ is designed based on (56), such that:

$$\|\tilde{\mathbf{x}}'_{k+1}\| < \|\mathbf{x}_k\|$$

It should be noted that such an input $\mathbf{u}_k$ will always exist since the system in Equation (52) is assumed point-wise controllable. The local truncation error (between the true nonlinear system and the discretized pseudo-linear system subject to the ZOH control law) at the end of the time interval is defined as:

$$\mathbf{e}_{k+1} = \tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_{k+1} \quad (57)$$

Define a term $\lambda(\mathbf{x}_k, \Delta t)$ as:

$$\lambda(\mathbf{x}_k, \Delta t) = \frac{\|\mathbf{e}_{k+1}\|}{\|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\|}$$

By taking the limit of $\lambda(\cdot, \Delta t)$ when $\Delta t \rightarrow 0$, the following is obtained:

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \lambda(\mathbf{x}_k, \Delta t) &= \lim_{\Delta t \rightarrow 0} \frac{\|\tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_{k+1}\|}{\|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\|} \\ &= \lim_{\Delta t \rightarrow 0} \frac{\|\tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_{k+1}\|}{\Delta t} \frac{1}{\frac{\|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\|}{\Delta t}} \end{aligned}$$

Let:

$$\lambda_F(\mathbf{x}_k, \Delta t) = \frac{\|\tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_{k+1}\|}{\Delta t}, \quad \lambda_G(\mathbf{x}_k, \Delta t) = \frac{\|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\|}{\Delta t}$$

One can then obtain:

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \lambda_F(\mathbf{x}_k, \Delta t) &= \lim_{\Delta t \rightarrow 0} \frac{\|\tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_{k+1}\|}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{\|(\tilde{\mathbf{x}}'_{k+1} - \mathbf{x}_k) - (\mathbf{x}_{k+1} - \mathbf{x}_k)\|}{\Delta t} \rightarrow 0 \\ \lim_{\Delta t \rightarrow 0} \lambda_G(\mathbf{x}_k, \Delta t) &= \lim_{\Delta t \rightarrow 0} \frac{\|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\|}{\Delta t} = \frac{\mathbf{x}_k^T (\mathbf{f}(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k) \mathbf{u}_k)}{\|\mathbf{x}_k\|} \end{aligned}$$

$\mathbf{x}_k \neq \mathbf{0} \Rightarrow \mathbf{f}(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k) \mathbf{u}_k \neq \mathbf{0}$. A stabilizing $\mathbf{u}_k$ dictates $\mathbf{x}_k^T (\mathbf{f}(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k) \mathbf{u}_k) \neq 0$. From all of this, it can be concluded that:

$$\lim_{\Delta t \rightarrow 0} \lambda(\mathbf{x}_k, \Delta t) = \frac{\lim_{\Delta t \rightarrow 0} \lambda_F(\mathbf{x}_k, \Delta t)}{\lim_{\Delta t \rightarrow 0} \lambda_G(\mathbf{x}_k, \Delta t)} \rightarrow \frac{0}{\frac{\mathbf{x}_k^T (\mathbf{f}(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k) \mathbf{u}_k)}{\|\mathbf{x}_k\|}} = 0$$

Thus, $\exists \Delta t \in (0, \infty)$, which ensures the corresponding $\lambda(\mathbf{x}_k, \Delta t) \in (0, 1)$. This leads to:

$$\|\mathbf{e}_{k+1}\| = \lambda(\mathbf{x}_k, \Delta t) \|\tilde{\mathbf{x}}'_{k+1}\| - \|\mathbf{x}_k\| < \|\mathbf{x}_k\| - \|\tilde{\mathbf{x}}'_{k+1}\|$$

Combined with Equation (57), one can then easily deduce that $\|\mathbf{x}_{k+1}\| < \|\mathbf{x}_k\|$ which means that the system would progress successively as, $\|\mathbf{x}_{i+1}\| < \|\mathbf{x}_i\| \quad i \rightarrow 0, 1, 2, \dots$. Therefore, one can conclude that $\exists \Delta t \in [0, \infty)$, which guarantees that a control law designed based on Equation (54) will stabilize the original system described by Equation (51). Thus, it is shown that the sampled-data ZOH controller based on the pseudo-linear representation in Equation (52) also ensures that the states of the true nonlinear system stay close to the states evolving based on Equation (52). The next section focuses on the constrained sampled-data NMPC based on the SDC system.

5.2. Stability of Constrained Sampled-Data NMPC Based on SDC System

The only difference between the constrained sampled-data NMPC and constrained linear MPC is that the feasibility of the system needs to be checked at the start of every sampling interval. For simplicity, the brackets will be replaced as subscripts to indicate that the dependence on $\mathbf{x}_k, (\cdot)(\mathbf{x}_k)$ would be written as $(\cdot)_k$. Similarly for each $\mathbf{x}_k$, there must exist a set of $(\mathbf{Q}_{fk}, \mathbf{Q}_k, \mathbf{R}_k, \mathbf{K}_k, \Delta \mathbf{U}_k)$ that satisfy the following conditions:

$$\gamma_k < \infty \quad (58)$$

$$\left[\begin{array}{c} \gamma_k - 2 \left[(\tilde{\mathbf{C}}_k^T \mathbf{F}_k \mathbf{x}_k)^T \tilde{\mathbf{Q}}_{Nk} \tilde{\mathbf{C}}_k \mathbf{H}_k + (\mathbf{C}_k \mathbf{A}_k^N \mathbf{x}_k)^T \mathbf{Q}_{fk} \mathbf{C}_k \tilde{\mathbf{B}}_k \right] \Delta \mathbf{U}_k \\ \Delta \mathbf{U}_k \end{array} \quad \begin{array}{c} \Delta \mathbf{U}_k^T \\ (\mathbf{H}_k^T \tilde{\mathbf{C}}_k^T \tilde{\mathbf{Q}}_{Nk} \tilde{\mathbf{C}}_k \mathbf{H}_k + \tilde{\mathbf{R}}_{Nk} + \tilde{\mathbf{B}}_k^T \mathbf{C}_k^T \mathbf{Q}_{fk} \mathbf{C}_k \tilde{\mathbf{B}}_k)^{-1} \end{array} \right] \geq 0 \quad (59)$$

$$\mathbf{\Gamma}(\mathbf{x}_k) \Delta \mathbf{U}_k \leq \mathbf{Y}(\mathbf{x}_k) \quad (60)$$

$$\begin{bmatrix} (C_k^T Q_{fk} C_k)^{-1} & (C_k^T Q_{fk} C_k)^{-1} (A_k - B_k K_k)^T & (C_k^T Q_{fk} C_k)^{-1} \sqrt{C_k^T Q_k C_k} & (C_k^T Q_{fk} C_k)^{-1} K_k^T \sqrt{R_k} \\ (A_k - B_k K_k)(C_k^T Q_{fk} C_k)^{-1} & (C_k^T Q_{fk} C_k)^{-1} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \\ \sqrt{C_k^T Q_k C_k} (C_k^T Q_{fk} C_k)^{-1} & \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} & \mathbf{0}_{n \times n} \\ \sqrt{R_k} K_k (C_k^T Q_{fk} C_k)^{-1} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} \end{bmatrix} \geq \mathbf{0} \quad (61)$$

$$\begin{bmatrix} \mathbf{I}_{m \times m} \\ -\mathbf{I}_{m \times m} \end{bmatrix} K_k (A_k^N x_k + \bar{B}_k \Delta U_k) \leq \begin{bmatrix} (u_{ub} - u_T) \\ -(u_{lb} - u_T) \end{bmatrix} \quad (62)$$

$$\begin{bmatrix} C_{zk} \\ -C_{zk} \end{bmatrix} (A_k - B_k K_k) [A_k^N x_k + \bar{B}_k (u_T + \Delta U_k)] \leq \begin{bmatrix} z_{ub} \\ -z_{lb} \end{bmatrix} \quad (63)$$

After obtaining a suitable (Q_{fk}, Q_k, R_k) from the above LMI, ΔU_k^* is obtained by doing the following optimization:

$$\Delta U_k^* = \arg \min_{\Delta U_k} \gamma_k$$

subject to the conditions specified in Equations (58)–(63). Upon solving the optimization problem, Δu_k^* can be obtained by:

$$\Delta u_k^* = [\mathbf{I}_{m \times m} \ \mathbf{0}_{m \times m} \ \cdots \ \mathbf{0}_{m \times m}] \Delta U_k^* \quad (64)$$

6. Simulation Results

In the preceding sections, two methods for generating control laws to track trajectories for a quadrotor with state and input constraints were discussed. All of the simulations were performed on a Fujitsu America i7 Laptop with 8 GB RAM and a 2.8-GHz processor running MATLAB R2016a. For all of the cases considered, there was always a converged optimization solution. The simulation had a provision to bypass the optimization if it exceeded the default maximum number of function evaluations and continue to use the previous converged solution. However, such a scenario was not encountered for the cases considered. The parameters used in the simulations are described in Appendix A.

For the linear case, the six degrees of freedom equations of motion were linearized, and the MPC algorithm was implemented, which optimized the actuator effort based on the constraints imposed. For all of the simulation results, the initial equilibrium condition was that of a hover. The quadrotor model (5) is linearized at a certain hover position denoted as $x_T = [p^T \ \mathbf{0}_{1 \times 3} \ \mathbf{0}_{1 \times 3} \ \mathbf{0}_{1 \times 3}]^T$, in which $p \in \mathcal{R}^3$ is any constant position. The input u_T needed to maintain the hover position is $u_T = [\frac{mg}{4} \ \frac{mg}{4} \ \frac{mg}{4} \ \frac{mg}{4}]^T$. Furthermore,

$$A_c = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & A_{c\Theta} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}$$

where the $A_{c\Theta} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and B_c matrix is the same as in Equation (4). The linear MPC control actions are obtained using the matrices given above that optimize Equation (13) subject to constraints Equation (21).

For the nonlinear case, the control laws are derived using the system matrices described in Section 4 that optimize the cost in Equation (48) subject to the constraints in Equation (50).

For both cases, the true system dynamics presented in Equation (51) along with the first order actuator dynamics for the rotor thrusts is simulated. Three different reference trajectories are generated to evaluate the effectiveness of the control laws: (a) helical, (b) spherical spiral and (c) straight line segments. The control input bounds are set as $u_{lb} = [0 \ 0 \ 0 \ 0]^T$ and $u_{ub} = [mg \ mg \ mg \ mg]^T$. The control horizon is chosen as $N = 25$.

Figure 2 shows the reference trajectory and the tracking errors for both the linear and the nonlinear MPC cases. It can be seen that both controllers achieve good tracking; however, the transient response of the nonlinear MPC is better. Clearly, from Figure 2b, the speed of response with regards to tracking the position errors is much better for the nonlinear MPC.

Figure 3 shows the results of tracking a spherical spiral trajectory. As with the previous case, both controllers achieve successful tracking.

Figure 4 shows the results of tracking a straight line trajectory segment between two way points. As with the previous cases, both controllers achieve successful tracking. In this case, too, the superior transient performance of the nonlinear MPC is clearly seen. Additional studies show that the nonlinear MPC is able to track the trajectories

even with a shorter prediction horizon. Table 1 showed the average root mean square (RMS) of position tracking errors of different prediction horizons. It can be seen that NMPC has better tracking performances in terms of position tracking. Table 2 showed the average control effort used in each tracking scenarios for different control horizons. In all three cases, NMPC is able to achieve better tracking with less control effort.

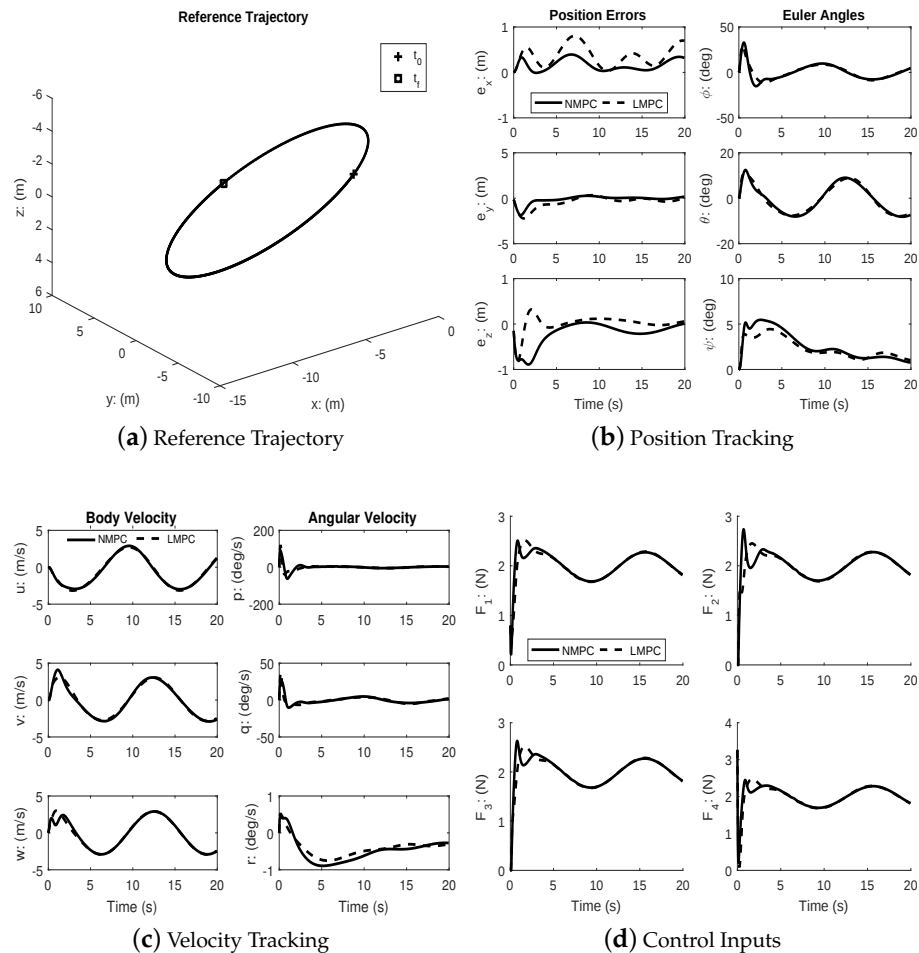


Figure 2. Helical trajectory tracking.

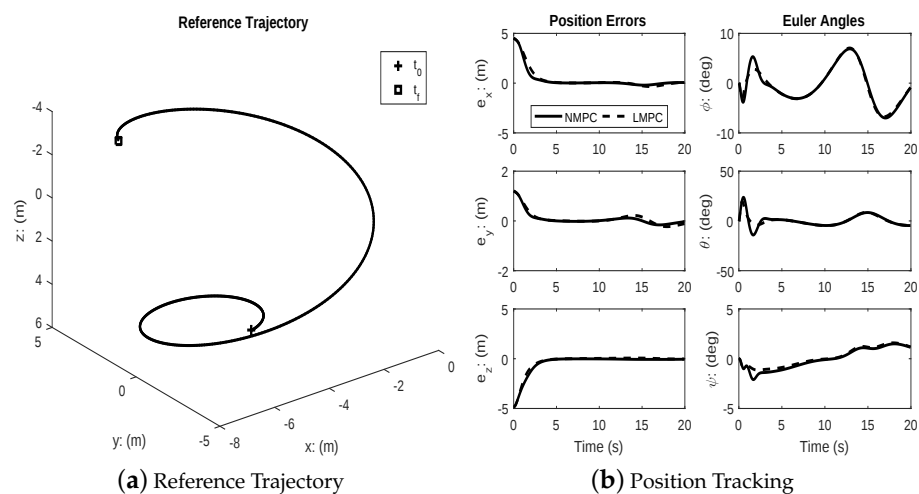


Figure 3. Cont.

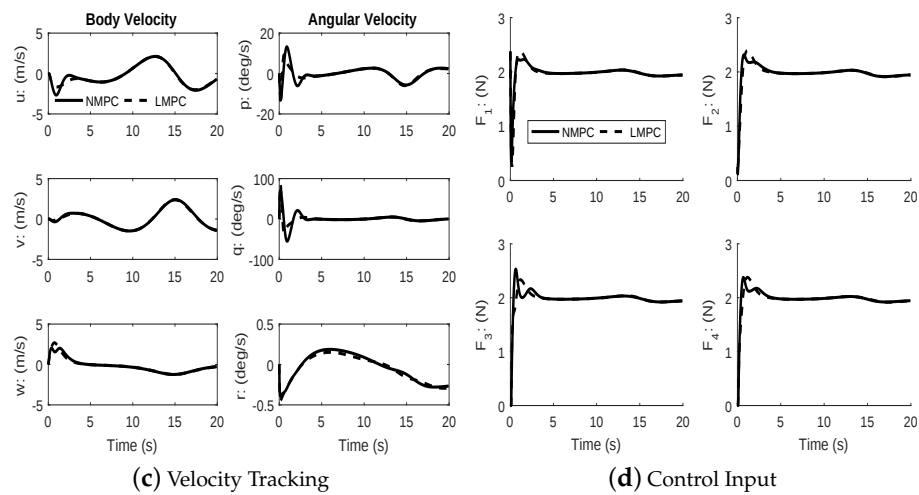


Figure 3. Spherical spiral trajectory.

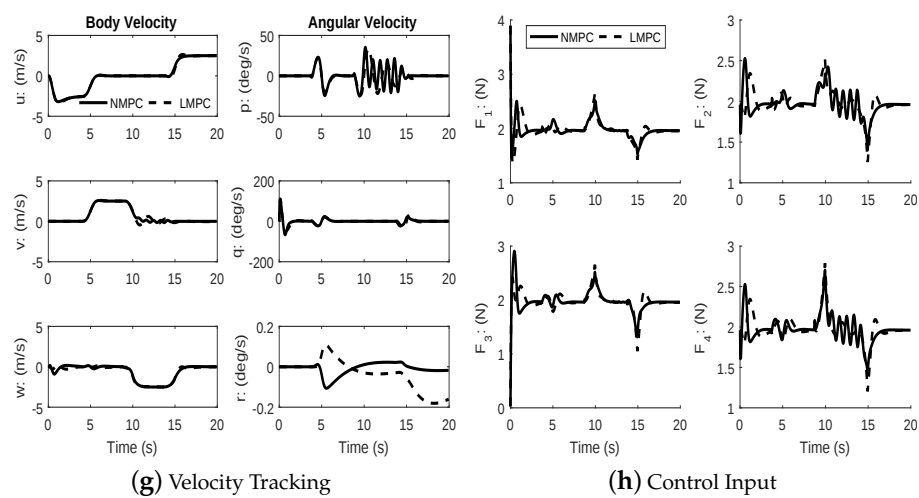
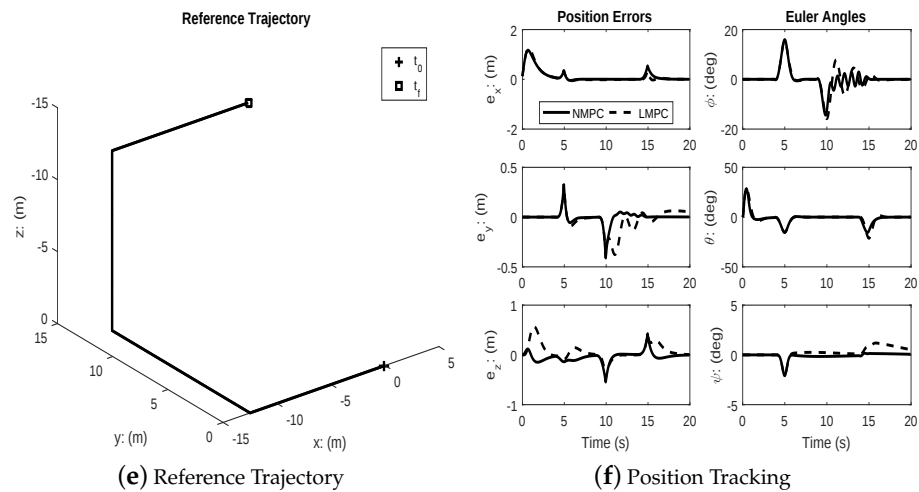


Figure 4. Straight line trajectory segments.

Table 1. RMS of position tracking errors.

Trajectory	Nonlinear/Linear	N = 10	N = 15	N = 25	N = 35	N = 45
Helical	Nonlinear	0.5663	0.5824	0.5831	0.5757	0.5832
Helical	Linear	0.8075	0.8348	0.8484	0.8364	0.8334
Spherical Spiral	Nonlinear	1.4739	1.4725	1.4725	1.4720	1.4726
Spherical Spiral	Linear	1.4984	1.5000	1.5016	1.5009	1.5009
Straight Line	Nonlinear	0.3112	0.3020	0.3169	0.3160	0.3155
Straight Line	Linear	0.3722	0.3575	0.3633	0.3612	0.3603

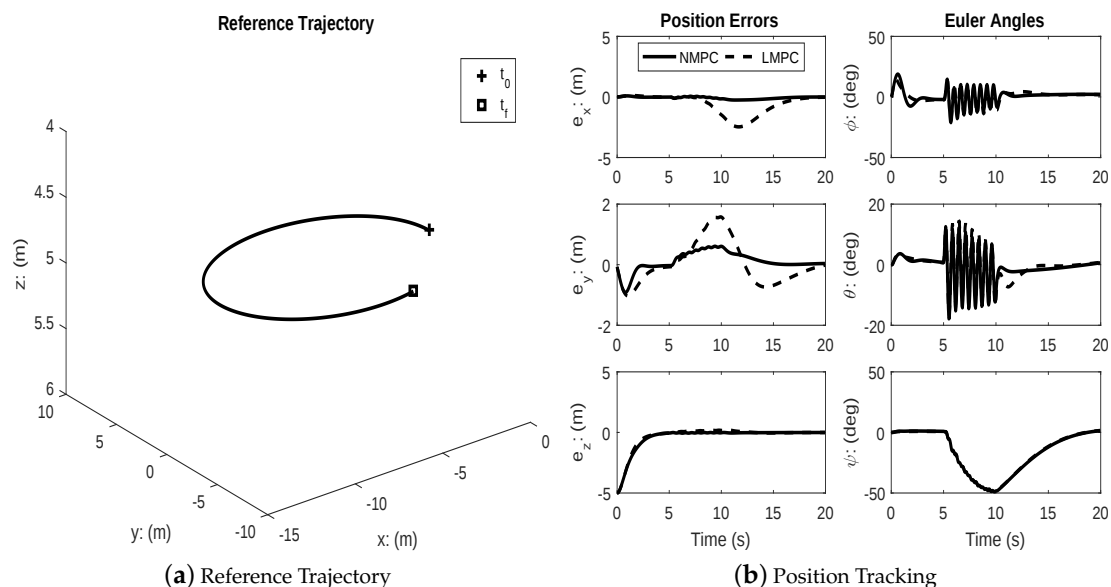
Table 2. Average control effort ($\Delta \mathbf{u}^T \Delta \mathbf{u}$).

Trajectory	Nonlinear/Linear	N = 10	N = 15	N = 25	N = 35	N = 45
Helical	Nonlinear	0.3838	0.3724	0.3652	0.3665	0.3674
Helical	Linear	0.4205	0.4107	0.4057	0.4084	0.4114
Spherical Spiral	Nonlinear	0.1687	0.1689	0.1700	0.1707	0.1710
Spherical Spiral	Linear	0.1910	0.1913	0.1914	0.1913	0.1913
Straight Line	Nonlinear	0.2125	0.1433	0.1282	0.1280	0.1296
Straight Line	Linear	0.2245	0.1593	0.1388	0.1379	0.1367

Nonlinear Model Predictive Control with Disturbance

In order to further assess the effectiveness of the NMPC scheme, simulations were performed with simultaneous high frequency disturbance torques injected in the roll and pitch channels. The disturbances are two exponentially decaying sinusoidal functions: $\delta \tau = e^{-0.1t}[\sin(10t); \cos(10t); 0]$ during $t \in (5, 10)(s)$. Figures 5–7 show the performance of linear and nonlinear MPC subject to disturbance. The disturbance is used to approximate the effects of wind during typical outdoor flight conditions.

Under disturbances, linear MPC and nonlinear MPC both performed well with an acceptable amount of tracking errors and kept the input within bounds. It can be seen from the figures that the nonlinear MPC significantly out-performs the linear MPC with regards to transient performance.

**Figure 5.** Cont.

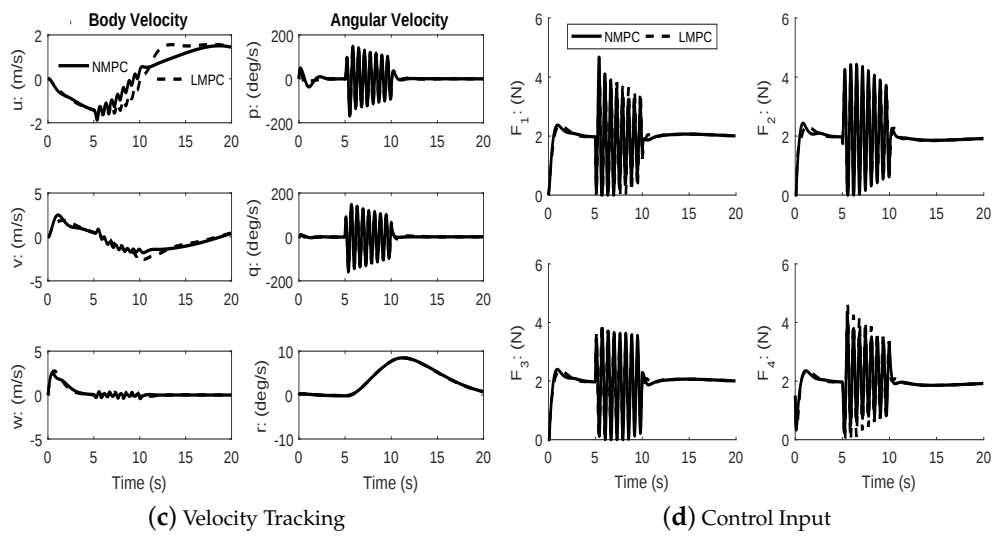


Figure 5. Helical trajectory (with disturbance).

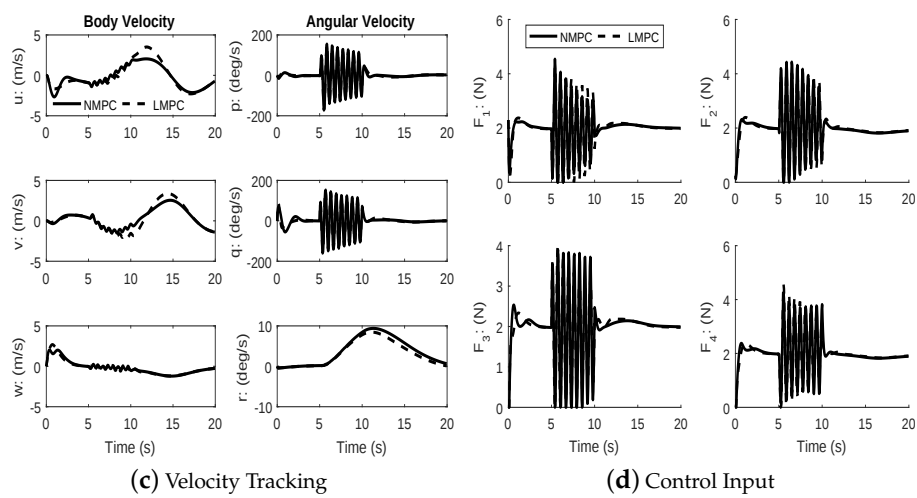
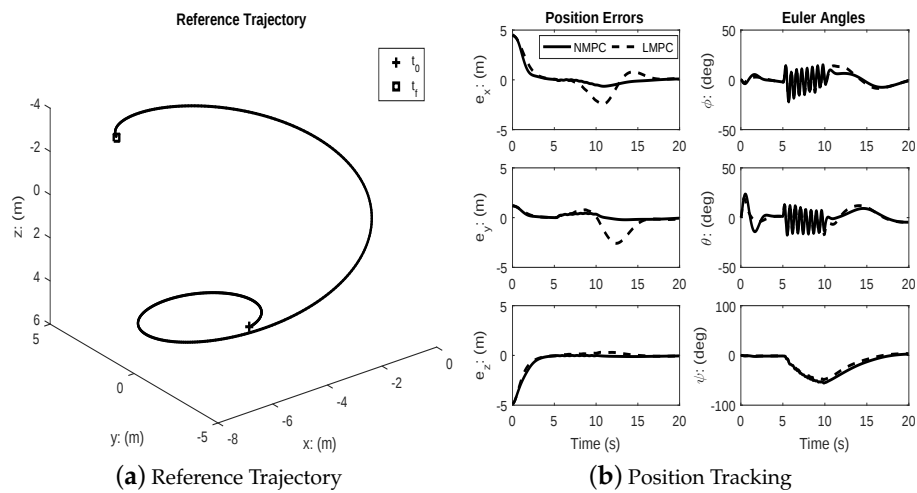


Figure 6. Spherical spiral trajectory (with disturbance).

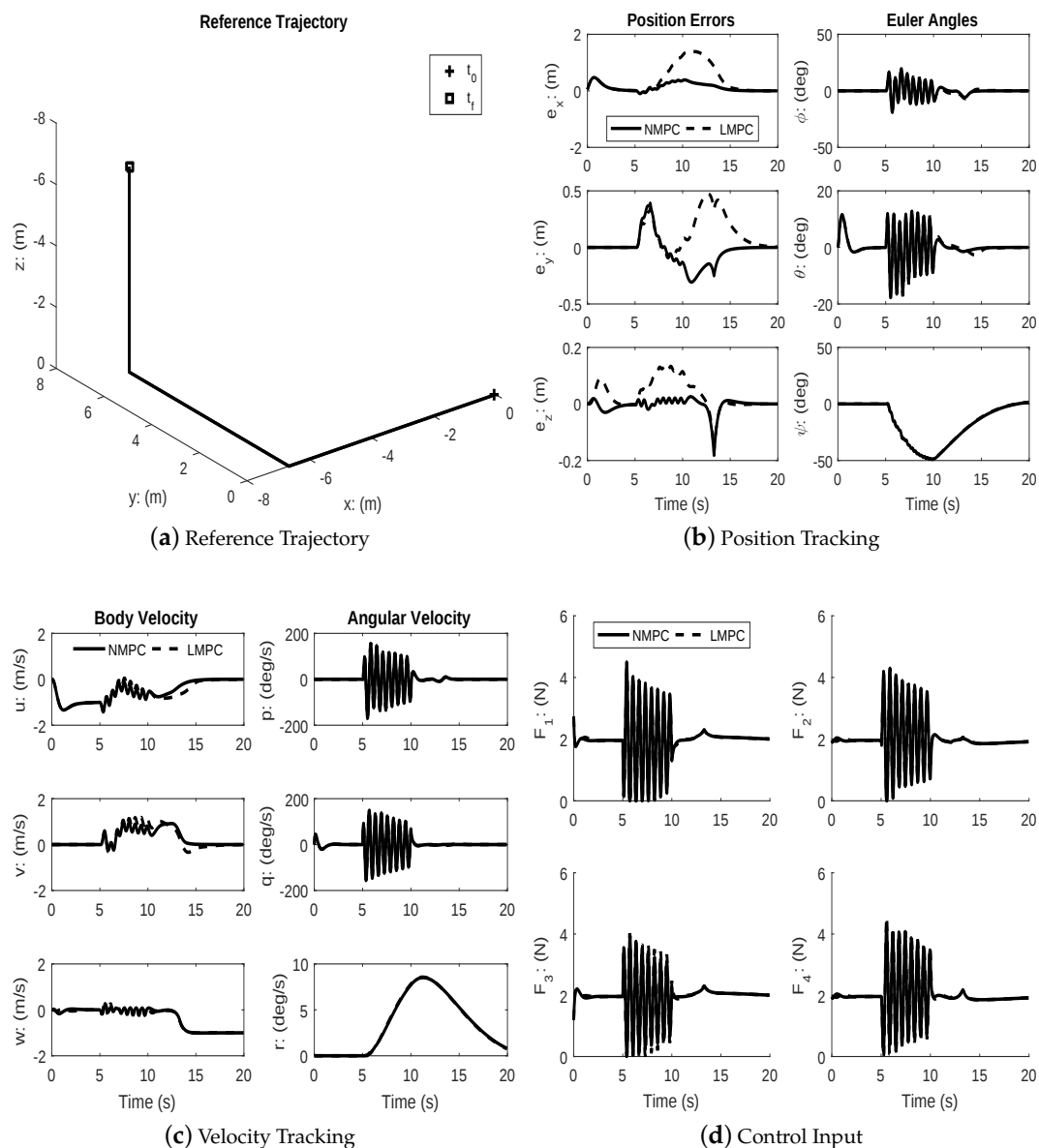


Figure 7. Straight line trajectory segments (with disturbance).

7. Summary and Conclusions

A linear and nonlinear model predictive control scheme were presented to control a quadrotor platform with an objective to track various reference trajectories. The nonlinear model predictive controller made use of a state-dependent coefficient representation of the nonlinear dynamics that accounted for the peculiarities of the platform under consideration. The closed loop stability analysis of the linear and nonlinear control schemes were presented to show the similarities, as well as to highlight the differences. The nonlinear controller clearly outperformed the linear controller with superior transient performance in both nominal conditions and in the presence of high frequency disturbance torques in the roll and pitch channels.

Author Contributions: Both Pengkai Ru and Kamesh Subbarao designed and performed the experiments, analyzed the data, and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A Simulation Parameters

The following parameters were used for all of the simulations. The parameters for the control law synthesis were the same regardless of the reference trajectories and the linear/nonlinear control schemes.

- Mass: $m = 0.8$ (kg).
- Moment of inertia: $\mathbf{J} = \begin{bmatrix} 0.0224 & & \\ & 0.0224 & \\ & & 0.0436 \end{bmatrix}$ ($\text{kg} \cdot \text{m}^2$).
- Distance from the center of the rotor to CG of the quadrotor: $L = 0.165$ (m).
- Ratio of rotor angular momentum to rotor lift: $c = 0.002167$ (m).
- Gravitational acceleration: $g = 9.8$ (m/s^2).
- Rotor actuator time constant: $\lambda_F = 50$.
- Control horizon: $N = 25$.
- Prediction horizon: $M = 25$.
- Sample time: $\Delta t = 0.05$ (s).
- State weighting matrix: $\mathbf{Q} = \text{diag}[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.
- Input weighting matrix: $\mathbf{R} = \text{diag}[1, 1, 1, 1]$.

References

1. Huang, H.; Hoffmann, G.M.; Waslander, S.L.; Tomlin, C.J. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '09, Kobe, Japan, 12–17 May 2009; pp. 3277–3282.
2. Materna, R. *Highlights from Aerospace Industry Report 2012: Facts, Figures and Outlook for the Aviation and Aerospace Manufacturing Industry*; Aerospace Industries Association of America Inc.: Arlington, VA, USA, 2012.
3. Augugliaro, F.; Lupashin, S.; Hamer, M.; Male, C.; Hehn, M.; Mueller, M.W.; Willmann, J.S.; Gramazio, F.; Kohler, M.; D'Andrea, R. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Syst.* **2014**, *34*, 46–64.
4. Lupashin, S.; Hehn, M.; Mueller, M.W.; Schoellig, A.P.; Sherback, M.; Andrea, R.D. A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics* **2014**, *24*, 41–54.
5. Lindsey, Q.; Mellinger, D.; Kumar, V. Construction with quadrotor teams. *Auton. Robots* **2012**, *33*, 323–336.
6. deVries, E.; Subbarao, K. Cooperative control of swarms of unmanned aerial vehicles. In Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Orlando, FL, USA, 4–7 January 2011; Number AIAA 2011-78.
7. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the International Conference on Robotics and Automation (ICRA 11), Shanghai, China, 9–11 May 2011.
8. Turpin, M.; Michael, N.; Kumar, V. Trajectory design and control for aggressive formation flight with quadrotors. *Auton. Robots* **2012**, *33*, 143–156.
9. Mellinger, D.; Michael, N.; Kumar, V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* **2012**, *31*, 664–674.
10. Subbarao, K.; Tule, C.; Ru, P. Nonlinear model predictive control applied to trajectory tracking for unmanned aerial vehicles. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA Aviation Forum, Dallas, TX, USA, 22–26 June 2015; Number AIAA 2015-2857.
11. Pivtoraiko, M.; Mellinger, D.; Kumar, V. Incremental micro-UAV motion replanning for exploring unknown environments. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 2452–2458.
12. De Luca, A.; Lanari, L.; Oriolo, G. A sensitivity approach to optimal spline robot trajectories. *Automatica* **1991**, *27*, 535–539.
13. De Luca, A.; Oriolo, G. Trajectory planning and control for planar robots with passive Last joint. *Int. J. Robot. Res.* **2002**, *21*, 575–590.
14. Bouktir, Y.; Haddad, M.; Chettibi, T. Trajectory planning for a quadrotor helicopter. In Proceedings of the 16th Mediterranean Conference on Control and Automation, Ajaccio, France, 25–27 June 2008; pp. 1258–1263.
15. Van Nieuwstadt, M.; Murray, R.M. Real time trajectory generation for differentially flat systems. *Int. J. Robust Nonlinear Control* **1996**, *8*, 995–1020.

16. Gillula, J.H.; Huang, H.; Vitus, M.P.; Tomlin, C.J. Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 1649–1654.
17. Hoffmann, G.M.; Waslander, S.L.; Tomlin, C.J. Quadrotor helicopter trajectory tracking control. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Honolulu, HI, USA, 18–21 August 2008.
18. Hehn, M.; D’Andrea, R. Quadrocopter trajectory generation and control. *World Congress* **2011**, *18*, 1485–1491.
19. Hehn, M.; Ritz, R.; D’Andrea, R. Performance benchmarking of quadrotor systems using time-optimal control. *Auton. Robots* **2012**, *33*, 69–88.
20. Palunko, I.; Fierro, R.; Cruz, P. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 2691–2697.
21. Bertrand, S.; Hamel, T.; Piet-Lahanier, H. Stability analysis of an UAV controller using singular perturbation theory. *IFAC Proc.* **2008**, *41*, 5706–5711.
22. Flores, G.; Lozano, R. Lyapunov-based controller using singular perturbation theory: An application on a mini-UAV. In Proceedings of the American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 1596–1601.
23. Das, A.; Subbarao, K.; Lewis, F.L. Dynamic inversion with zero-dynamics stabilization for quadrotor control. *IET Control Theory Appl.*, **2009**, *3*, 303–314.
24. deVries, E.; Subbarao, K. Backstepping based nested multi-loop control laws for a quadrotor. In Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision, Singapore, 7–10 December 2010.
25. Schoellig, A.P.; Mueller, F.L.; D’Andrea, R. Optimization-based iterative learning for precise quadrocopter trajectory tracking. *Auton. Robots* **2012**, *33*, 103–127.
26. Das, A.; Lewis, F.L.; Subbarao, K. Backstepping approach for controlling a quadrotor using neural networks. *J. Intell. Robot. Syst.* **2009**, *56*, 127–151.
27. Kim, H.J.; Shim, D.H.; Sastry, S. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In Proceedings of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002; Volume 5, pp. 3576–3581.
28. Pamadi, B.N. *Performance, Stability, Dynamics, and Control of Airplanes*; AIAA: Reston, VA, USA, 2004; p. 320.
29. Lupashin, S.; Schöllig, A.; Sherback, M.; D’Andrea, R. A simple learning strategy for high-speed quadrocopter multi-flips. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 1642–1648.
30. Lopes, R.V.; Santana, P.H.R.Q.A.; Borges, G.A.; Ishihara, J.Y. Model predictive control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft. In Proceedings of the COBEM, 21st International Congress of Mechanical Engineering, Natal, RN, Brazil, 24–28 October 2011.
31. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814.
32. Alexis, K.; Nikolakopoulos, G.; Tzes, A. Model predictive control scheme for the autonomous flight of an unmanned quadrotor. In Proceedings of the 2011 IEEE International Symposium on Industrial Electronics (ISIE), Gdansk, Poland, 27–30 June 2011; pp. 2243–2248.
33. Findeisen, R.; Raff, T.; Allgower, F. Sampled-data model predictive control for constrained continuous time systems. In *Advanced Strategies in Control Systems with Input and Output Constraints*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 207–235.
34. Wills, A.; Heath, W.P. Application of barrier function model predictive control to an edible oil refining process. *J. Process Control* **2005**, *15*, 183–200.
35. Camacho, E.F.; Bordons, C. *Model Predictive Control*; Springer: London, UK, 1999; pp. 18–20.
36. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004; pp. 152–156.
37. Kwon, W.H.; Han, S.H. *Receding Horizon Control: Model Predictive Control for State Models*; Springer Science & Business Media: London, UK, 2006.
38. Lewis, F.L.; Syrmos, V.L. *Optimal Control*; Wiley-Interscience: New York, NY, USA, 1995.

39. Cimen, T. State-Dependent Riccati Equation (SDRE) control: A survey. In Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC), Seoul, Korea, 6–11 July 2008; Volume 17, pp. 3761–3775.
40. Bogdanov, A.; Carlsson, M.; Harvey, G.; Hunt, J.; Kieburz, D.; Merwe, R.V.D.; Wan, E. State-Dependent Riccati Equation control of a small unmanned helicopter. In Proceedings of the AIAA Guidance Navigation and Control Conference, Austin, TX, USA, 11–14 August 2003.
41. Hammett, K. Control of Nonlinear Systems via State Feedback State-Dependent Riccati Equation Techniques. Ph.D. Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, OH, USA, 1997.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).