EE372 2021

Dynamic Programming and Optimal Control

Final Project

NMPC-based control for Quadrotor trajectory tracking subject to input constraints

Student: Yi-Hsuan Chen

Advisor: Taous-Meriem Laleg-Kirati

MS student, Department of Mechanical Engineering

King Abdullah University of Science and Technology (KAUST)

May 2021

Contents

Chapter 1

Introduction

Quadrotors have been used in a wide variety of fields such as exploration, surveillance, transportation and even entertainment in the past few decades due to versatility and affordability.The higher maneuverability and stability is easily achieved compared to fixed-wing aircraft and conventional helicopter due to the ability to perform a vertical takeoff and landing (VTOL). Since Quadrotors become more and more popular, numerous control methods have been developed for both regulation and trajectory tracking. The objective is to find a control strategy that allows the states of a quadrotor to converge to an arbitrary set of time-varying reference states. Though it is possible to control a quadrotor using linear control techniques by linearizing the system around a trim point even if the quadrotor model is coupled and highly nonlinear, nonlinear control methods are preferred to obtain better performance. Multiple nonlinear methods such as sliding mode, backstepping, adaptive and feedback linearization have been demonstrated to be effective for quadrotor control. However, none of these control methods take state and control constraints into account, we might face instability issues due to saturation and violation of feasible region. Motivated by this point, the nonlinear MPC is proposed to control the quadrotor and handle input constraints at the same time. Also, less effort for tuning is required in MPC structure, since we can simply put weights on states based on the degree of importance. The main drawback of MPC is the computation complexity, because the optimization problem need to be solved iteratively over a whole time period. Thanks to the development of technology, more powerful solvers are available to cope with MPC's computation demand.

The work is organized as follows. In Chapter 2 the mathematical model of a quadrotor is derived. MPC is introduced in Chapter 3 with the reformulation optimal control problem as a nonlinear programming problem. In Chapter 4 the two numerical simulations are demonstrated to analyze the performance of the NMPC strategy. Finally, conclusions and future work are drawn in Chapter 5.

Chapter 2

Mathematical model of quadcopter

The quadrotor configuration is shown in Fig. 2.1, which includes the corresponding angular velocities, forces and torques generated by the four rotors.

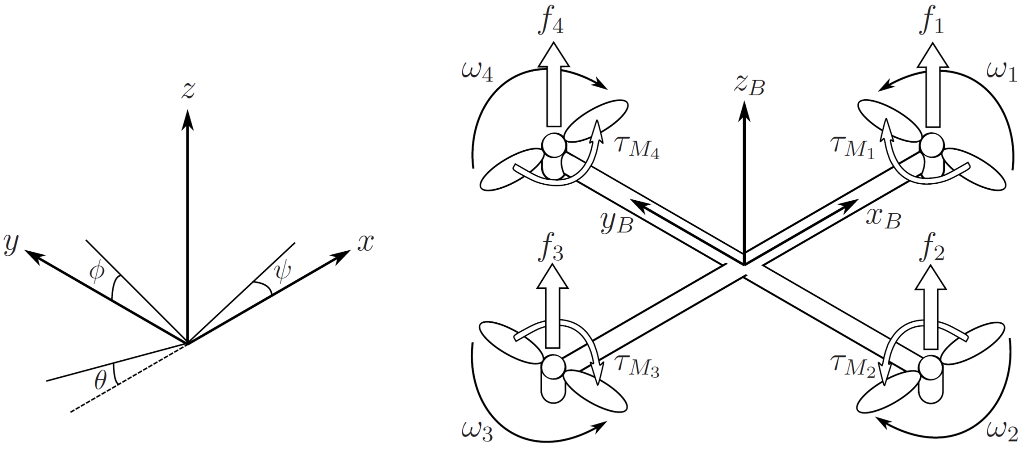


Figure 2.1: The inertial and body frames of a quadcopter [1].

The absolute linear position of the quadrotor is defined in the inertial frame x, y, z axes with $\boldsymbol{\xi}$. The angular position, i.e., Euler angles is defined in the inertial frame with three Euler angles $\boldsymbol{\eta}$. Pitch angle θ, roll angle ϕ and yaw angle ψ determine the rotations around y-axis, x-axis and z-axis, respectively. Vector $\mathbf{q}$ consists of the position and attitude vectors.

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix}, \tag{2.1}$$

The origin of the body frame is in the center of mass of the quadcopter, while body z-axis z_B is pointing vertically upward and the body x-axis x_B is pointing to rotor 1. In the body frame, the translational velocities are defined as $\mathbf{V}_B$ and the angular velocities as $\boldsymbol{\omega}$

$$\mathbf{V}_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.2}$$

The rotation matrix from the body frame to the inertia frame is

$$\mathbf{R}(\boldsymbol{\eta}) = \begin{bmatrix} C_\psi C_\theta & -S_\psi C_\phi + C_\psi S_\theta S_\phi & S_\psi S_\phi + C_\psi S_\theta C_\phi \\ S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{2.3}$$

where $S_x = \sin(x)$ and $C_x = \cos x$. All of the transformation matrices, including $\mathbf{R}(\boldsymbol{\eta})$, are orthonormal, which means their inverse is equivalent to their transpose, that is, $\mathbf{R}^{-1} = \mathbf{R}^T$, where $\mathbf{R}^{-1}$ is the rotation matrix from the inertial frame to the body frame. The quadcopter is assumed to be symmetric with the four arms aligned with the body x- and y-axes. Thus, the diagonal inertia matrix $\mathbf{I}$ is

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.4}$$

The total forces of rotors create thrust T in the direction of the body z-axis. Torque $\boldsymbol{\tau}_B$ is consisted of the torques τ_ϕ, τ_θ and τ_ψ in the direction of the corresponding body frame angles.

$$\mathbf{T}_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 f_i \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{\tau}_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} (-f_2 + f_4)\ell \\ (-f_1 + f_3)\ell \\ (-f_1 + f_2 - f_3 + f_4)\mu \end{bmatrix} \tag{2.6}$$

where ℓ is the distance between the rotor and the center of mass of the quadrotor and μ is the drag constant. The relationship between external forces, torques and propeller thrusts is

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\ell & 0 & \ell \\ \ell & 0 & -\ell & 0 \\ -\mu & \mu & -\mu & \mu \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{2.7}$$

2.1 Euler-Lagrange equations

The two approaches to build the mathematical model of a quadrotor are Newton-Euler method and Euler-Lagrange method. In this study, the Lagrange method is adopted to derive quadrotor equations of motion. A scalar function called the Lagrangian $\mathcal{L}$ is introduced, which represents the difference between the total kinetic energy $\mathcal{T}$ and the total potential energy $\mathcal{U}$ of the system. The total kinetic energy contains the translational and rotational energies.

$$\begin{aligned} \mathcal{L} &= \mathcal{T} - \mathcal{U} = T_{\text{trans}} + T_{\text{rot}} - \mathcal{U} \\ &= \frac{1}{2} m \dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} + \frac{1}{2} \boldsymbol{\omega}^T \mathbf{I} \boldsymbol{\omega} - mgz \end{aligned} \tag{2.8}$$

The Euler-Lagrange equations with generalized forces and torques is written as

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{f}_B \\ \boldsymbol{\tau}_B \end{bmatrix} \tag{2.9}$$

The external linear force $\mathbf{f}_B$ is the total force acted in the inertial frame. The linear Euler-Lagrange equations are

$$\mathbf{f}_B = \mathbf{R}(\boldsymbol{\eta})\mathbf{T}_B = m\ddot{\boldsymbol{\xi}} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.10}$$

The transformation matrix for angular velocities from the inertial frame to the body frame is $\mathbf{\Phi}$.

$$\boldsymbol{\omega} = \mathbf{\Phi}\dot{\boldsymbol{\eta}}, \qquad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.11}$$

Define the Jacobian matrix $\mathbf{J}(\boldsymbol{\eta})$ from $\boldsymbol{\omega}$ to $\dot{\boldsymbol{\eta}}$ as

$$\mathbf{J}(\boldsymbol{\eta}) = \mathbf{J} = \mathbf{\Phi}^T\mathbf{I}\mathbf{\Phi}, \tag{2.12}$$

Rewrite the rotational energy in the inertia frame as

$$T_{\text{rot}} = \frac{1}{2}\boldsymbol{\omega}^T\mathbf{I}\boldsymbol{\omega} = \frac{1}{2}\dot{\boldsymbol{\eta}}^T\mathbf{J}\dot{\boldsymbol{\eta}} \tag{2.13}$$

The external angular force $\boldsymbol{\tau}_B$ is the torques of the rotors. The angular Euler-Lagrange equations are

$$\boldsymbol{\tau}_B = \mathbf{J}\ddot{\boldsymbol{\eta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\eta}} - \frac{1}{2}\frac{\partial}{\partial\boldsymbol{\eta}}\left(\dot{\boldsymbol{\eta}}^T\dot{\boldsymbol{\eta}}\right) = \mathbf{J}\ddot{\boldsymbol{\eta}} + \mathbf{C}\left(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}\right) \tag{2.14}$$

where the matrix $\mathbf{C}\left(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}\right)$ is the Coriolis term, including the gyroscopic and centripetal terms. The matrix $\mathbf{C}\left(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}\right)$ is shown in [2].

$$\mathbf{C}\left(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}\right) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$$
\begin{aligned}
C_{11} &= 0 \\
C_{12} &= (I_y - I_z)(\dot{\theta} C_\phi S_\phi + \dot{\psi} S_\phi^2 C_\theta) + (I_z - I_y)\dot{\psi} C_\phi^2 C_\theta - I_x \dot{\psi} C_\theta \\
C_{13} &= (I_z - I_y)\dot{\psi} C_\phi S_\phi C_\theta^2 \\
C_{21} &= (I_z - I_y)(\dot{\theta} C_\phi S_\phi + \dot{\psi} S_\phi C_\theta) + (I_y - I_z)\dot{\psi} C_\phi^2 C_\theta + I_x \dot{\psi} C_\theta \\
C_{22} &= (I_z - I_y)\dot{\phi} C_\phi S_\phi \\
C_{23} &= -I_x \dot{\psi} S_\theta C_\theta + I_y \dot{\psi} S_\phi^2 S_\theta C_\theta + I_z \dot{\psi} C_\phi^2 S_\theta C_\theta \\
C_{31} &= (I_y - I_z)\dot{\psi} C_\theta^2 S_\phi C_\phi - I_x \dot{\theta} C_\theta \\
C_{32} &= (I_z - I_y)(\dot{\theta} C_\phi S_\phi S_\theta + \dot{\phi} S_\phi^2 C_\theta) + (I_y - I_z)\dot{\phi} C_\phi^2 C_\theta \\
&\quad + I_x \dot{\psi} S_\theta C_\theta - I_y \dot{\psi} S_\phi^2 S_\theta C_\theta - I_z \dot{\psi} C_\phi^2 S_\theta C_\theta \\
C_{33} &= (I_y - I_z)\dot{\phi} C_\phi S_\phi C_\theta^2 - I_y \dot{\theta} S_\phi^2 C_\theta S_\theta - I_z \dot{\theta} C_\phi^2 C_\theta S_\theta + I_x \dot{\theta} C_\theta S_\theta
\end{aligned}
\tag{2.15}
$$

Rearrange (2.14) yields the differential equations for the angular accelerations

$$\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-1}(\boldsymbol{\tau}_B - \mathbf{C}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}\right)) \tag{2.16}$$

Rewrite the system into a compact form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.17}$$

where $\mathbf{x}$ denotes state variables $(\boldsymbol{\xi}, \boldsymbol{\eta}, \dot{\boldsymbol{\xi}}, \dot{\boldsymbol{\eta}})^T$ and $\mathbf{u}$ denotes control input $(f_1, f_2, f_3, f_4)^T$.

Chapter 3

Model Predictive Control and Nonlinear Programming

3.1 What is MPC?

Model predictive control (MPC), as known as receding horizon control (RHC), is an iterative process of optimizing the predictions of states in a finite horizon.

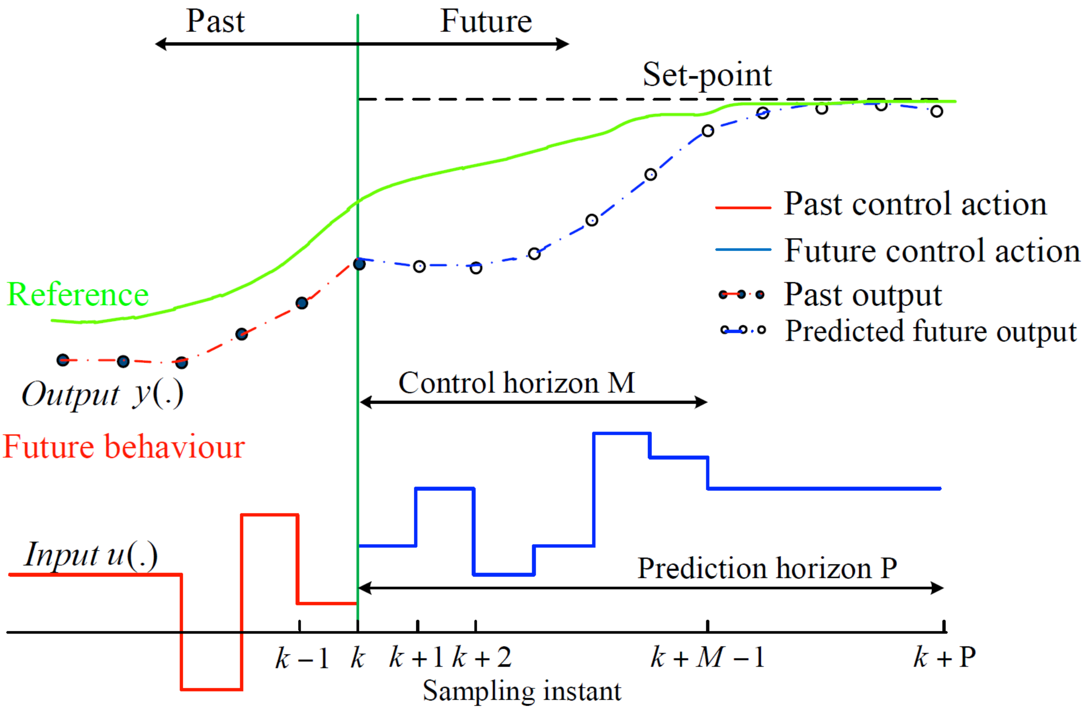


Figure 3.1: A general MPC structure [3].

Current time step measurements and future predicted outputs are used to track the reference over the horizon. The future prediction is based on the knowledge of system model. At current time step k, the optimal control sequence over the horizon is calculated by minimizing the cost function. Only the first step of the control sequence is implemented, then the plant state is measured again and the optimization is repeated starting from the updated current state, yielding a new control action and new predicted state trajectory. The prediction horizon keeps being shifted forward and can therefore be called as a receding horizon. In conclusion, the three main components in MPC strategy are prediction using system model, online optimization and receding implementation.

3.2 Mathematical Formulation of MPC

First, a running costs $\ell(\mathbf{x}, \mathbf{u})$ is defined to characterizes the control objective, which generally involves the difference between the measured and reference states and penalization of control actions.

$$\ell(\mathbf{x}, \mathbf{u}) = \|\mathbf{x}_{\mathbf{u}} - \mathbf{x}_r\|_{\mathbf{Q}}^2 + \|\mathbf{u} - \mathbf{u}_r\|_{\mathbf{R}}^2 \tag{3.1}$$

where where $\mathbf{Q}$ and $\mathbf{R}$ are the weight matrices specifying the weights on tracking the reference states and penalizing the control input, respectively. The higher weight means that tracking a certain state or penalizing of an input is deemed more important.

The cost function J_N is the evaluation of the running costs along the whole horizon, which is minimized over the horizon.

$$J_N(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{N-1} \ell(\mathbf{x}_{\mathbf{u}}(k), \mathbf{u}(k)) \tag{3.2}$$

A general formulation [4] for an MPC optimal control problem (OCP) is

$$\begin{aligned}
\min_{\mathbf{u}} \quad & J_N(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} \ell(\mathbf{x}_{\mathbf{u}}(k), \mathbf{u}(k)) \\
\text{subject to} \quad & \mathbf{x}_{\mathbf{u}}(k+1) = \mathbf{f}(\mathbf{x}_{\mathbf{u}}(k), \mathbf{u}(k)) \\
& \mathbf{x}_{\mathbf{u}}(0) = \mathbf{x}_0 \\
& \mathbf{x}_{\mathbf{u}}(k) \in \mathcal{X}, \ \forall\, k \in [0, N] \\
& \mathbf{u}(k) \in \mathcal{U}, \ \forall\, k \in [0, N-1]
\end{aligned} \tag{3.3}$$

where the states and control inputs are within sets $\mathcal{X}$ and $\mathcal{U}$

$$\begin{aligned}
\mathcal{X} &:= \{x \in \mathbb{R}^n \,|\, \underline{x} \le x \le \bar{x}\} \\
\mathcal{U} &:= \{u \in \mathbb{R}^m \,|\, \underline{u} \le u \le \bar{u}\}
\end{aligned} \tag{3.4}$$

The current time step is t and the horizon is denoted by N. Notice that the optimization problem is subject to the system dynamics $\mathbf{x}_{\mathbf{u}}(k+1) = \mathbf{f}(\mathbf{x}_{\mathbf{u}}(k), \mathbf{u}(k))$.

3.3 Numerical Methods to Optimal Control Problem

In general, there are three major methods to solve optimal control problems [5], dynamic programming, indirect and direct methods Fig. 3.2.

(a) Dynamic Programming (DP) uses the principle of optimality to compute a feedback control recursively and to solve so called the Hamilton-Jacobi-Bellman (HJB) equation, a partial differential equation (PDE) in state space.

(b) Indirect methods use the necessary conditions of optimality of the infinite problem to derive a boundary value problem (BVP) in ordinary differential equations (ODE). The well-known calculus of variations, the Euler-Lagrange equations, and the Pontryagin Maximum Principle (PMP) are classified in this group.

(c) Direct methods cast the original optimal control problem (OCP) as a nonlinear programming problem (NLP). All direct methods are based on a finite dimensional parameterization of the control trajectory, but handle the state trajectory in different ways.

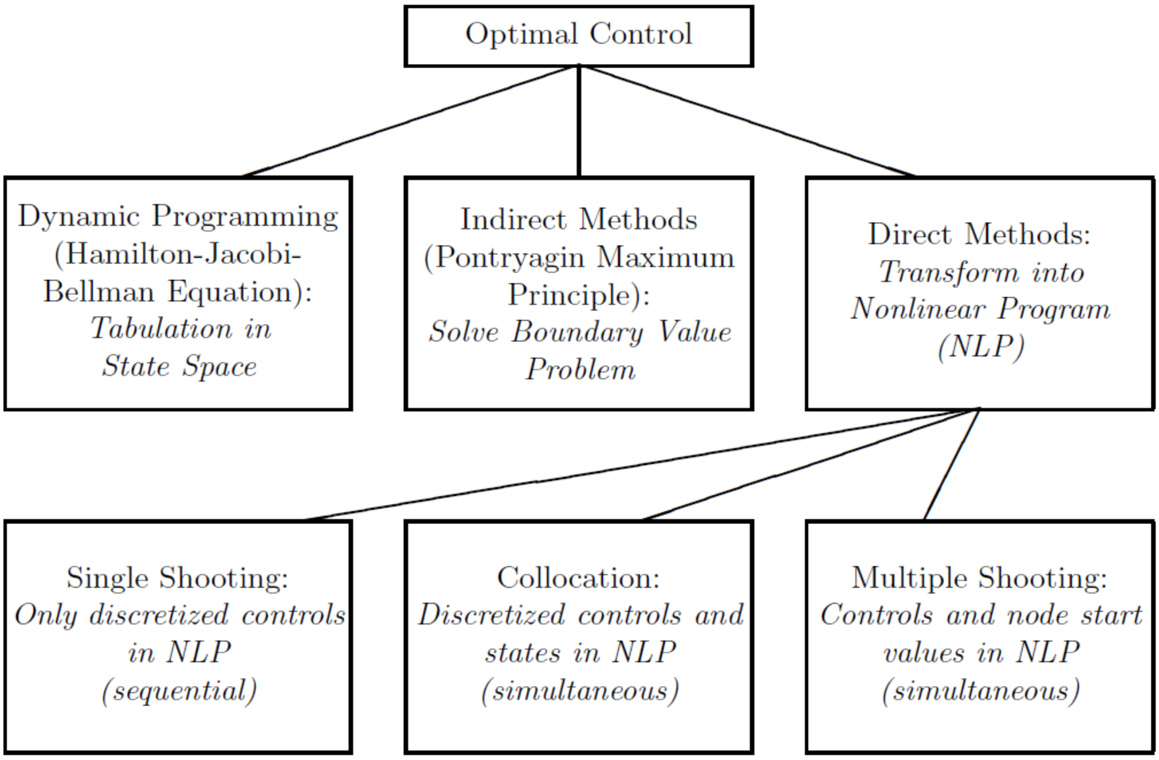

Optimal Control
Dynamic Programming (Hamilton-Jacobi-Bellman Equation): *Tabulation in State Space*
Indirect Methods (Pontryagin Maximum Principle): *Solve Boundary Value Problem*
Direct Methods: *Transform into Nonlinear Program (NLP)*
Single Shooting: *Only discretized controls in NLP (sequential)*
Collocation: *Discretized controls and states in NLP (simultaneous)*
Multiple Shooting: *Controls and node start values in NLP (simultaneous)*

Figure 3.2: Overview of numerical methods for optimal control [5].

3.3.1 Nonlinear Programming

For online NMPC the nonlinear programming problem must be solved numerically at every sampling interval, thus the optimal control problem (OCP) (3.4) needs to be formulated as a Nonlinear Programming problem (NLP) (3.5).

$$\begin{aligned} \min_{\mathbf{z}} \quad & f(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{z} \in \mathcal{Z} \\ & \mathbf{g}(\mathbf{z}) \leq 0 \\ & \mathbf{h}(\mathbf{z}) = 0 \end{aligned} \tag{3.5}$$

The optimization problem is rewritten as a function of a set of decision variables $\mathbf{z}$. The objective function $f(\mathbf{z})$ is the mapping from decision variables $\mathbf{z}$ to cost function J_N. Once NLP formulation is obtained, the NLP can be solved by existing NLP solvers, e.g. CVX, CasADi. In this study, a direct multiple shooting method was chosen to address NMPC, in which the model dynamics is discretized in a uniform time interval. Since a multiple shooting is more numerically stable and guarantees better convergence of NLPs typically.

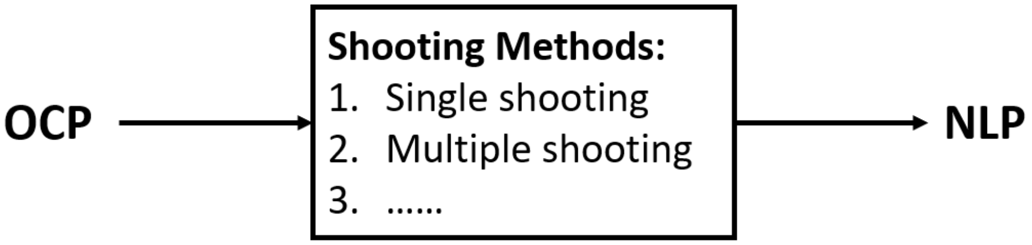
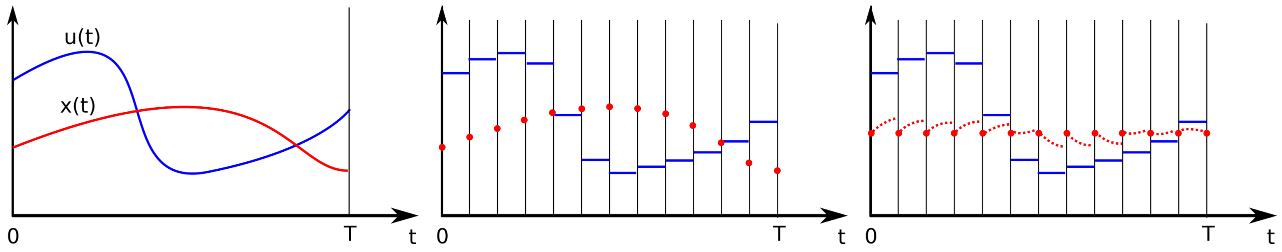


Figure 3.3: Relationship between OCP and NLP [6].

3.3.2 Implementation and Multiple Shooting

In multiple shooting, not only control variables $\mathbf{u}$ but also state variables $\mathbf{x}$ are treated as decision variables $\mathbf{z}$ within the NLP, in which system dynamics is imposed as equality condition to ensure continuity [4]. The state trajectory is discretized at the boundaries of each control interval and the control input is kept constant in each shooting interval, see Fig. 3.4(b). Now we have decision variables $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ and $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N+1}$. The start state $\mathbf{x}_k$ and a fixed control input $\mathbf{u}_k$ are known in each shooting interval k, so that the state at the next interval $\mathbf{x}_{k+1}$ can be predicted by numerical integration.



(a) Continuous state and control

(b) Discretized state and control

(c) Integrated state trajectories

Figure 3.4: Illustration of multiple shooting [7].

In Fig. 3.4(c), we found that there is a mismatch between where the integrator says we will end up and where our state decision variables think we are. Thus, continuity constraints are imposed to make the gap zero.

$$\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1} = 0 \tag{3.6}$$

A fourth-order Runge-Kutta approach is chosen to integrate ODE to obtain the state at the next time step $\mathbf{x}_{k+1}$ with sampling time h. An interior point method (IPM) is used to solve the NLP obtained from multiple shooting approach.

$$\begin{aligned}\hat{\mathbf{x}}_{k+1} &= \mathbf{x}_k + \frac{h}{6}(R_1 + 2R_2 + 2R_3 + R_4)\\ R_1 &= \mathbf{f}(t_k.\mathbf{x}_k)\\ R_2 &= \mathbf{f}(t_k + h/2, \mathbf{x}_k + R_1/2)\\ R_3 &= \mathbf{f}(t_k + h/2, \mathbf{x}_k + R_2/2)\\ R_4 &= \mathbf{f}(t_k + h, \mathbf{x}_k + R_3)\end{aligned} \tag{3.7}$$

The interior point method is applied using the Interior Point OPTimizer (IPOPT) solver in CasADi [6]. CasADi is an open-source tool for nonlinear optimization and algorithmic differentiation. It facilitates rapid yet efficient implementation of different methods for numerical optimal control, both in an offline context and for nonlinear model predictive control (NMPC) [8].

Chapter 4

Numerical Simulation

4.1 NMPC of a Quadrotor Control

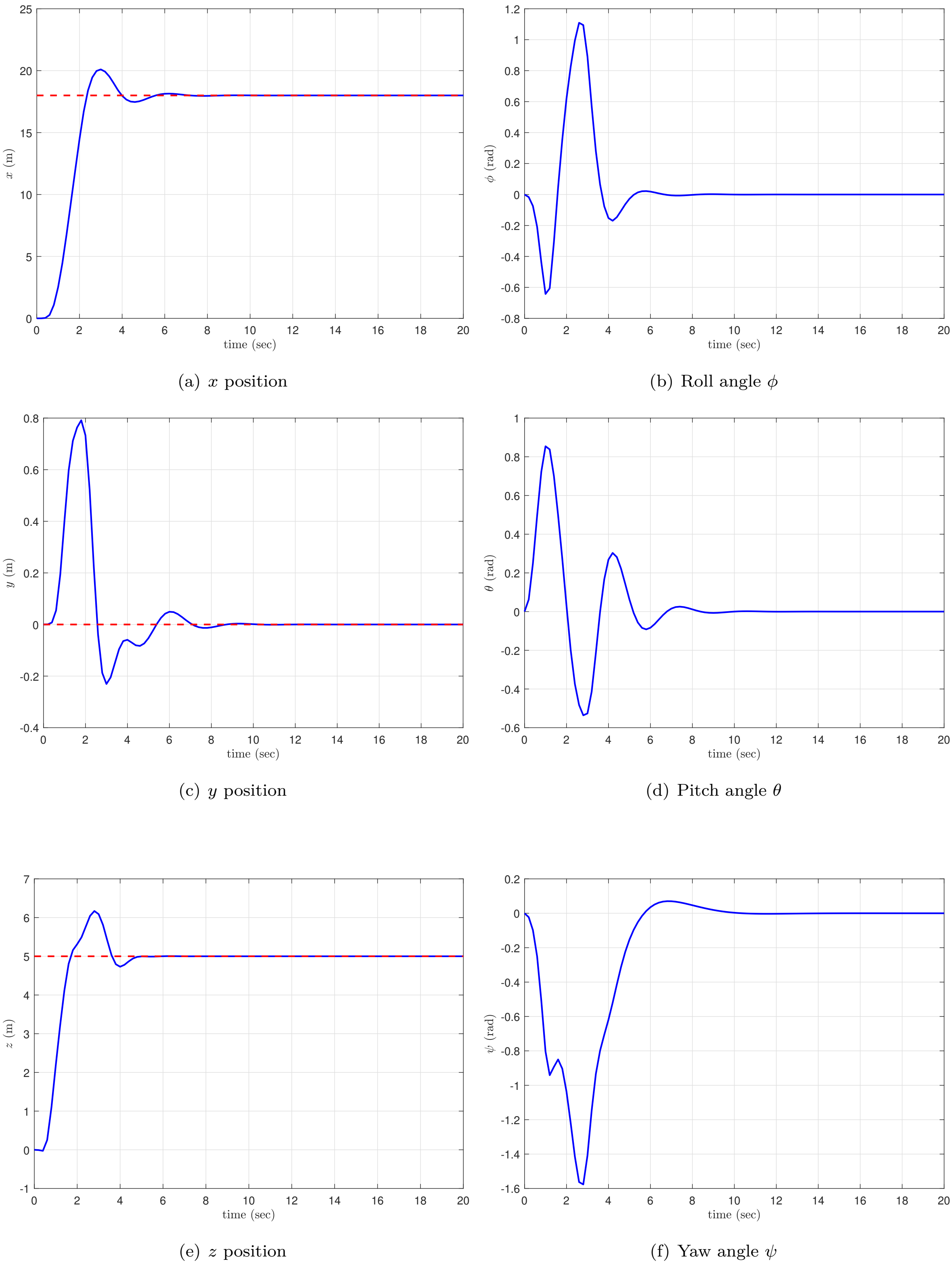
The dynamica model of a quadrotor is given by (2.10) and (2.16). The NMPC problem for quadrotor is formulated as

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_{k=0}^{N-1} \|\mathbf{x}_{\mathbf{u}} - \mathbf{x}_r\|_{\mathbf{Q}}^2 + \|\mathbf{u} - \mathbf{u}_r\|_{\mathbf{R}}^2 \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \\ & 0 \leq \mathbf{u}(k) \leq 15 \end{aligned} \tag{4.1}$$

The penalty on state error for x, y, z, ψ are 50 and 20. The penalty on state error for ϕ, θ and all control inputs are 1. The quadrotor parameters used in the simulation are given by: mass $m = 1\,kg$, gravity $g = 9.81\,m/s^2$, moment of inertia $I_x = I_y = 1.2\,kg \cdot m^2, I_z = 2\,kg \cdot m^2$, length of arm $\ell = 0.25\,m$, drag coefficient $\mu = 0.2$. The NMPC simulation parameters are as follows: prediction horizon $N = 30$ and sampling time $T_s = 0.1\,s$. We demonstrate two different flight missions, i.e., position stabilization and trajectory tracking, in the following sections.

4.1.1 Position Stabilization

The control objective of position stabilization is to steer the system from the origin to the target point $(x, y, z) = (18, 0, 5)$. The state and control trajectories are shown in Fig. 4.0 and Fig. 4.1, respectively.



(a) x position

(b) Roll angle ϕ

(c) y position

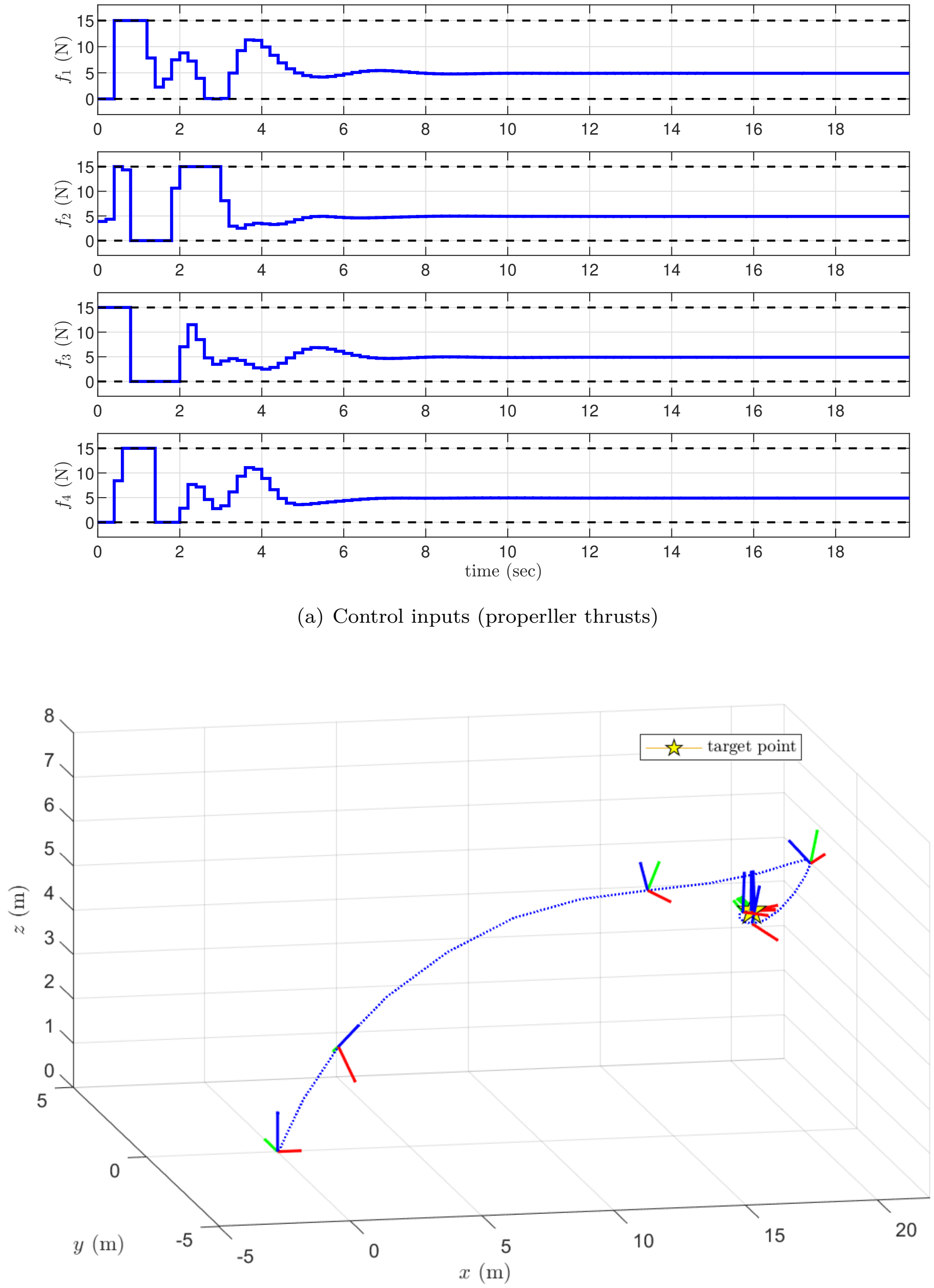
(d) Pitch angle θ

(e) z position

(f) Yaw angle ψ

Figure 4.0: State trajectories

It is found that the control inputs, i.e., propeller thrust, operate within limitation over the whole time, proving that NMPC is capable of handling constraints, as shown in Fig.4.1(a).



(a) Control inputs (properller thrusts)

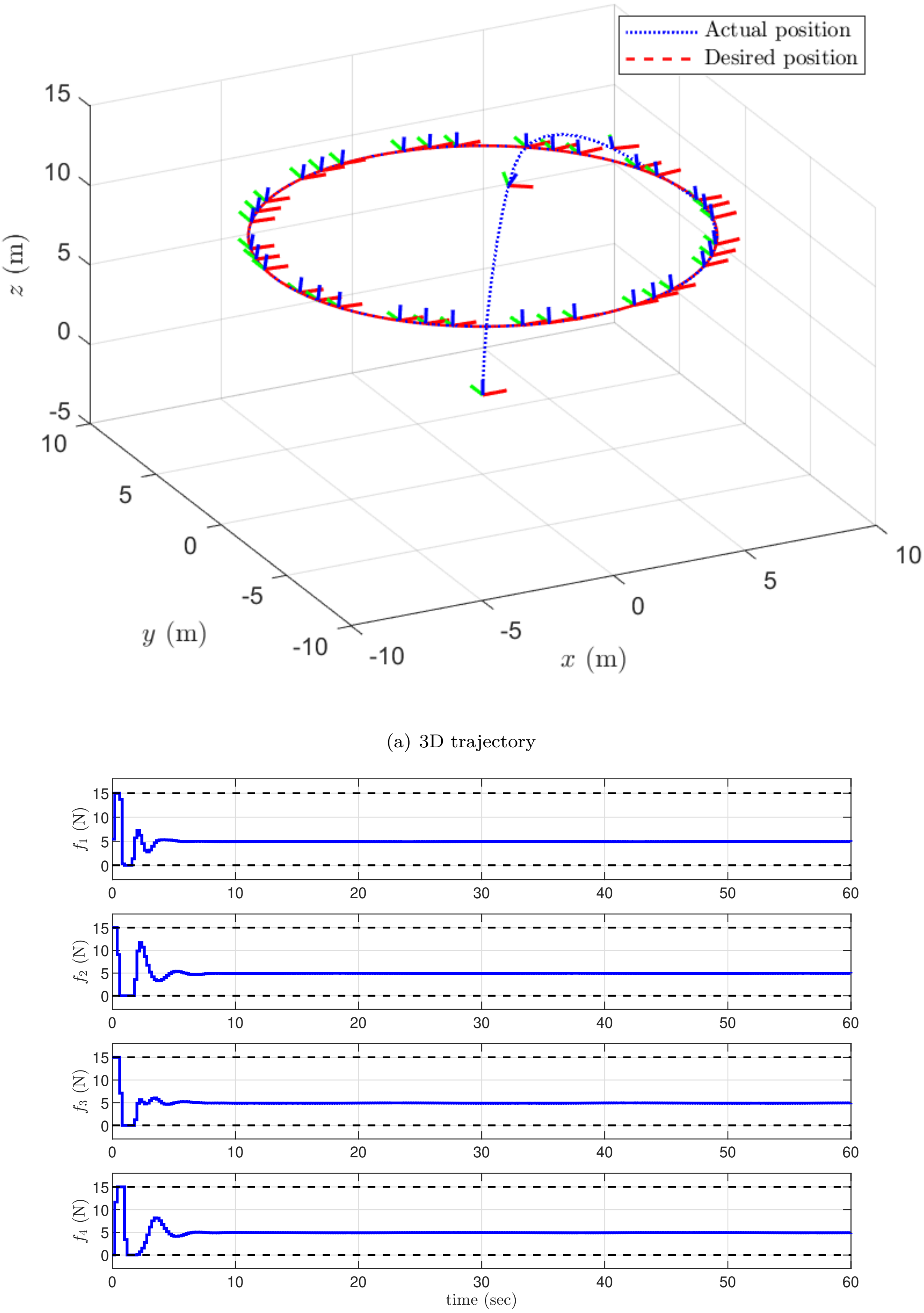
(b) 3D trajectory

Figure 4.1: Position Stablization

4.1.2 Trajectory Tracking

The control objective of trajectory tracking is to make the system follow the given time-varying trajectory. Here we introduced three different scenarios.

(i) Circular trajectory tracking: The quadrotor is required to be controlled along a circular path described as follows: $r = \begin{bmatrix} 8\sin(t/3) & 8\cos(t/3) & 10 \end{bmatrix}^T$.

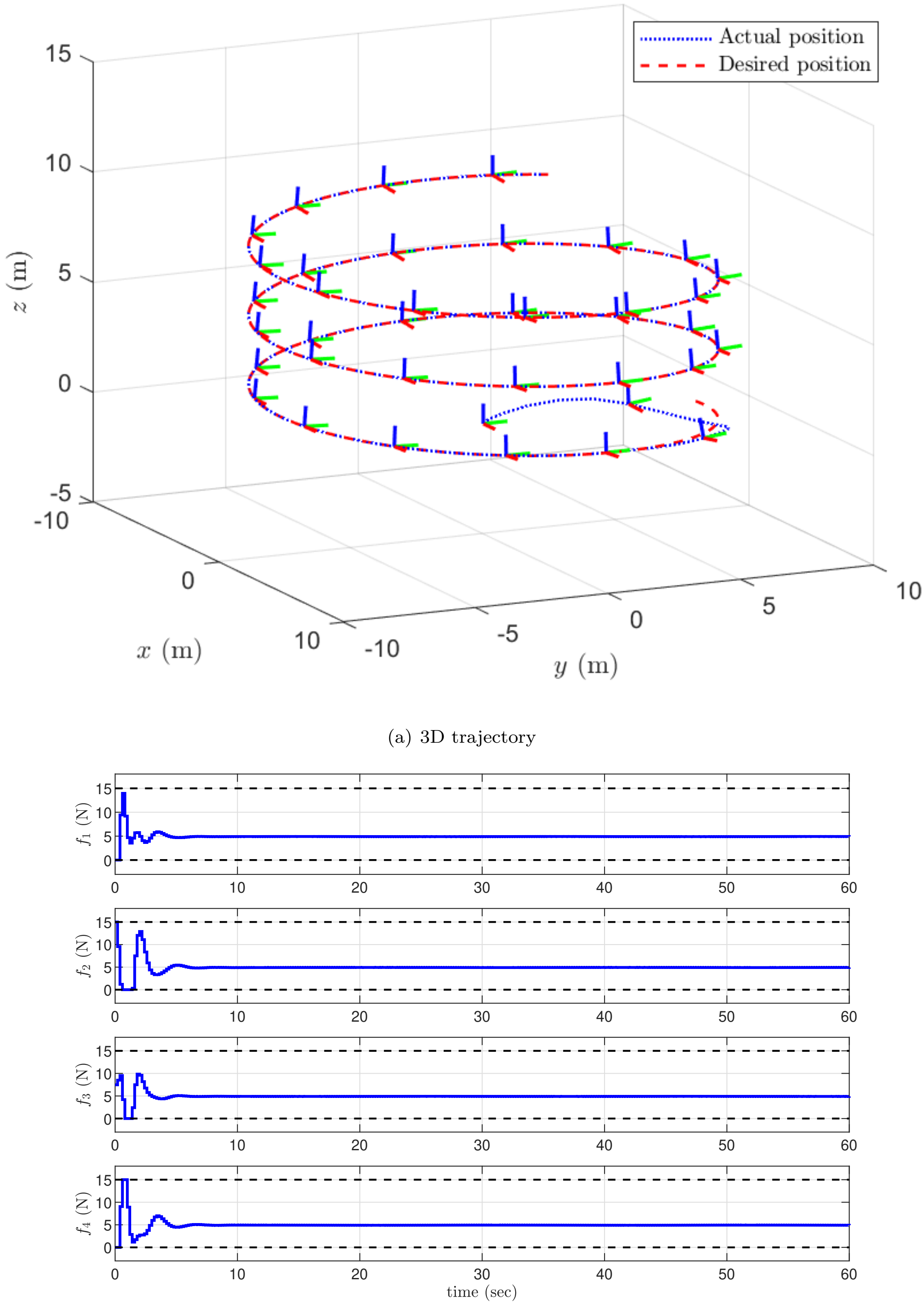


(a) 3D trajectory

(b) Control inputs (properller thrusts)

Figure 4.2: Circular trajectory tracking

(ii) Helix trajectory tracking: The quadrotor is required to be controlled along a helix path described as follows: $r = \begin{bmatrix} 8\sin(0.3t) & 8\cos(0.3t) & 0.15t \end{bmatrix}^T$.



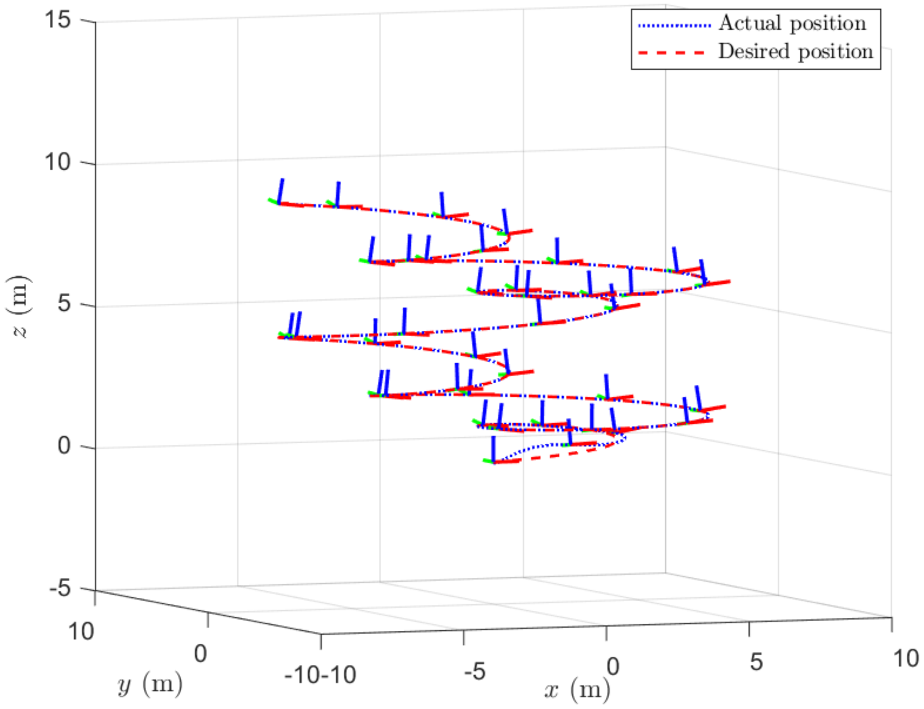
(a) 3D trajectory

(b) Control inputs (properller thrusts)

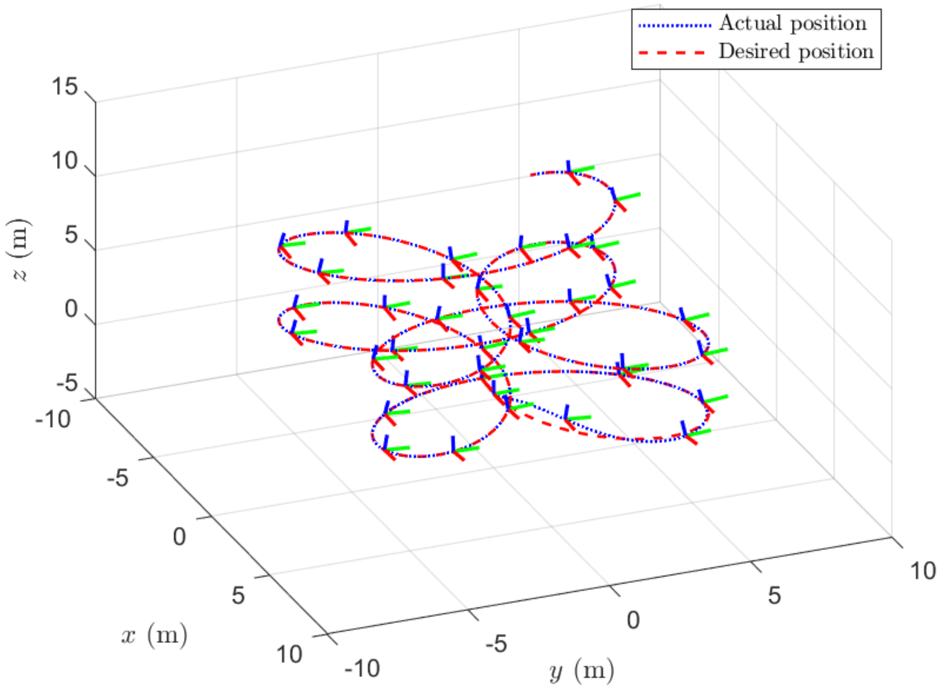
Figure 4.3: Helix trajectory tracking

(iii) Complex helix trajectory tracking: The quadrotor is required to be controlled along a helix path described as follows:

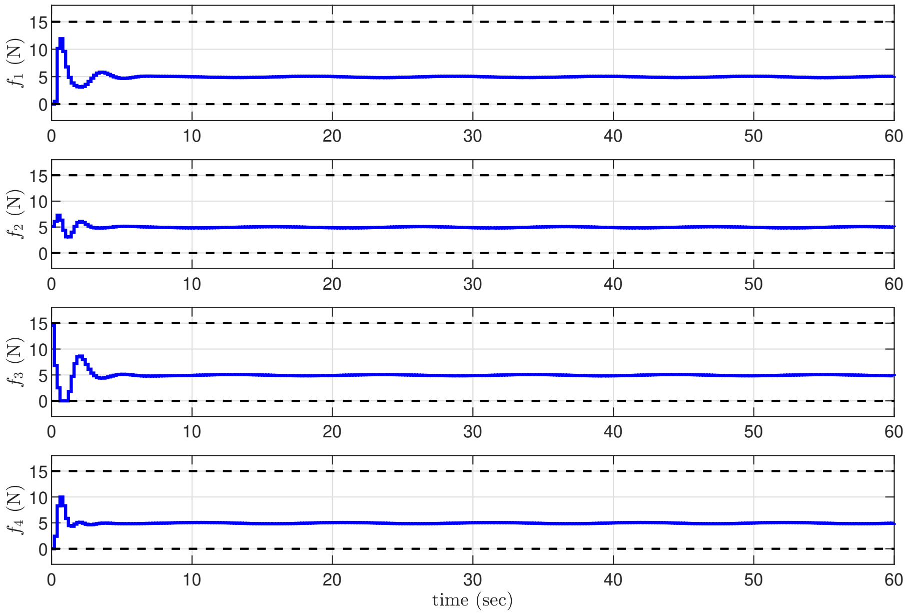
$$r = \begin{bmatrix} 15(\sin(0.2t) - (\sin(0.2t))^3) & 15(\cos(0.2t) - (\cos(0.2t))^3) & 0.15t \end{bmatrix}^T.$$



(a) 3D trajectory



(b) 3D trajectory



(c) Control inputs (properller thrusts)

Figure 4.4: Complex helix trajectory tracking

From above simulation results, as shown in Fig. 4.2, Fig. 4.3 and Fig. 4.4, the NMPC had the ability to stabilize the quadrotor on the desired trajectory, and to keep the control inputs in the operation range over a whole time interval simultaneously. The results demonstrated that the implemented MPC can achieve satisfactory performance and handle constraints, which is more practicable compared to conventional control methods.

Chapter 5

Conclusion

In this study, a nonlinear MPC controller is developed to achieve trajectory tracking for a quadrotor subject to input constraints. The optimal control problem is transformed into a nonlinear programming problem using multiple shooting method, which requires less computational effort and provides higher numerical stability. The IPOPT solver of the open-source optimal tool CasADi is used to solve the nonlinear programming problem in an efficient way. Multiple flight scenarios were performed to show the NMPC address the control problem in high performance.

5.1 Future Work

The closed-loop of the quadrotor was not guranteed in every desired trajectories. The stability issue involves the initial condition, selection of terminal constraints and length of prediction horizon. Therefore, more systematic stability analysis should be furthered investigated to improve reliability. Also, the model uncertainty is always existed in real world, the approach to alleviate the influence of model error should be studied.

Bibliography

[1] Teppo Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22:22, 2011.

[2] Guilherme V Raffo, Manuel G Ortega, and Francisco R Rubio. An integral predictive/nonlinear h∞ control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.

[3] Xiaoliang Yang, Guorong Liu, Anping Li, Le Van Dai, et al. A predictive power control strategy for dfigs based on a wind energy converter system. *Energies*, 10(8):1098, 2017.

[4] Gijs van Lookeren Campagne. A nonlinear model predictive control based evasive manoeuvre assist function. 2019.

[5] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.

[6] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. I. Mann, R. G. Gosine, and M. Diehl. Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. *IEEE Transactions on Control Systems Technology*, 24(4):1394–1406, 2016.

[7] CaSADi. Optimal control problems in a nutshell, 2018.

[8] CaSADi. Build efficient optimal control software, with minimal effort., 2018.