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

EE372 Progress Report

Yi-Hsuan Chen

Mechanical Engineering, MS student
King Abdullah University of Science and Technology (KAUST)

2020/05/03

Outline

Outline

Problem Statement

- Main Objective:
  Develop a model predictive controller to achieve trajectory tracking for a quadrotor with input constraints.
- Keywords:
  Quadrotor trajectory tracking Control, Multiple shooting, Nonlinear model predictive control, Input constraints

Outline

MPC Strategy Structure

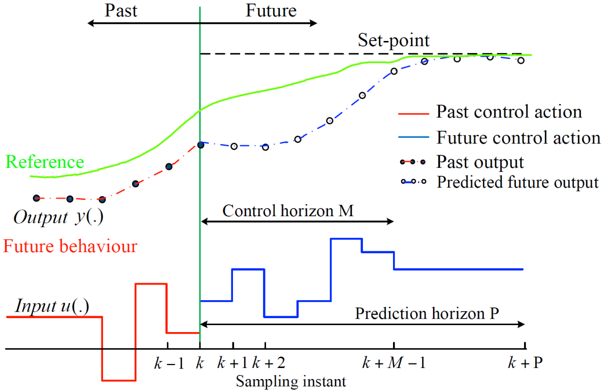
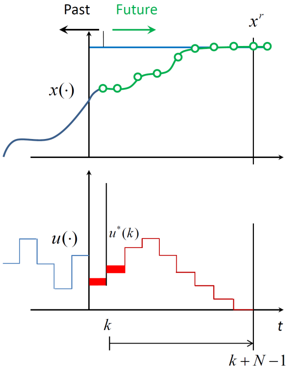


Figure: A general MPC structure [3].

MPC Strategy Structure

MPC Strategy Summary:

1. Prediction based on model
2. Online optimization
3. Receding horizon implementation



MPC Formulation

- **Running costs:** characterizes the control objective

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

  where **Q** and **R** are the weight matrices specifying the weights on tracking the reference states and penalizing the control input, respectively.
- **Cost function:** Evaluation of the running costs along the whole prediction horizon

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

MPC Formulation

- **Optimal Control Problem (OCP):** to find a minimizing control sequence

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

Outline

Quadrotor Dynamic Model

- state variables $\mathbf{x} = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})^T$
- control input $\mathbf{u} = (f_1, f_2, f_3, f_4)^T$

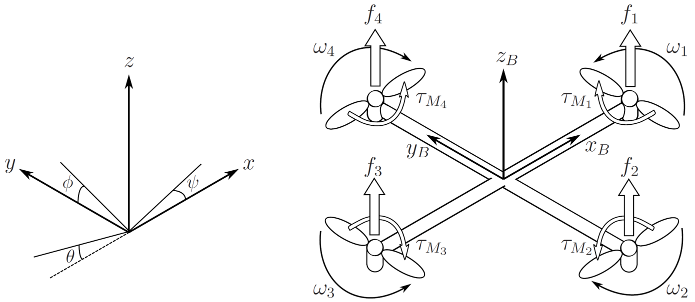


Figure: The inertial and body frames of a quadrotor

Quadrotor Dynamic Model

The equations of motion of a quadrotor are derived using Lagrange's method.

- Translational motion

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ \cos\theta\cos\phi \end{bmatrix} \frac{u_t}{m} + \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2}$$

  where u_t is total thrust force.

- Rotational motion $\eta = [\phi, \theta, \psi]^T$

$$\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-1}\left(\boldsymbol{\tau}_\eta - \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\right) \tag{3}$$

  where $\boldsymbol{\tau}_B = (\tau_\phi, \tau_\theta, \tau_\psi)$.

Quadrotor Dynamic Model

The external force and torques of a quadrotor are $T, \tau_\phi, \tau_\theta$ and τ_ψ representing thrust, roll, pitch, yaw torques, respectively, which can be expressed in terms of the propeller thrusts f_1, f_2, f_3, f_4.

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\ell & 0 & \ell \\ \ell & 0 & -\ell & 0 \\ -\mu & \mu & -\mu & \mu \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{4}$$

where ℓ is the distance between the rotor and the center of mass of the quadrotor and μ is the drag constant.

Quadrotor Dynamic Model

Rewrite the nonlinear equations (2) and (3) into a compact form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{5}$$

A Runga-Kutta method is used to obtain the state at the next time step $\mathbf{x}_{k+1}$ with sampling time h.

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

Overview of numerical methods for optimal control

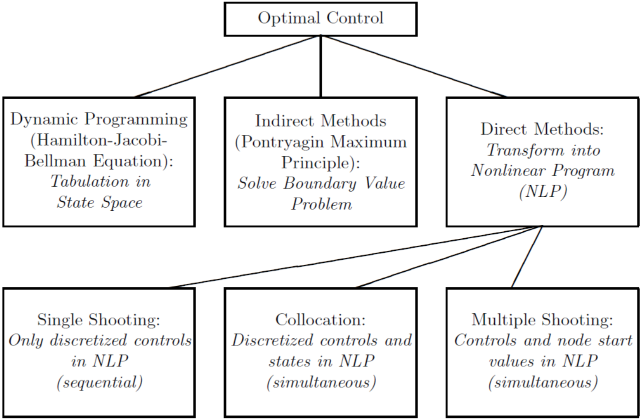


Figure: Numerical methods for optimal control [2].

OCP and NLP

To solve an NMPC problem, the optimization problem (1) needs to be reformulated as a Nonlinear Programming problem (NLP) (7).

$$\begin{aligned} \min_{\mathbf{z}} \quad & f(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{z} \in \mathcal{Z} \\ & g_i(\mathbf{z}) \leq 0 \ \text{ for each } i \in [1, ..., m] \\ & h_j(\mathbf{z}) = 0 \ \text{ for each } j \in [1, ..., p] \end{aligned} \tag{7}$$

The optimization problem is rewritten as a function of a set of decision variables $\mathbf{z}$. Once NLP formulation is obtained, the NLP can be solved by existing NLP solvers.

OCP and NLP

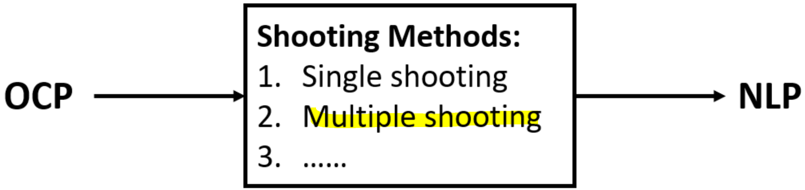
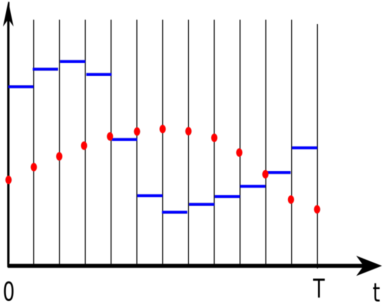
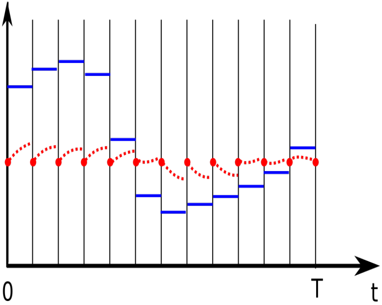


Figure: Relationship between OCP and NLP

Multiple Shooting Method



(a) State trajectory



(b) Control input

Figure: Illustration of multiple shooting [1].

Multiple Shooting Method - cont'd

- Not only control inpus **u** but also state variables **x** are decision variables in the optimization process.
- Additional continuity constraints are added to make sure that there is no mismatch between two states.

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

An interior point method is used to solve the NLP obtained from the multiple shooting approach. The open-source package CasADi is used to implement and solve the NLP.

Brief Introduction to CasADi

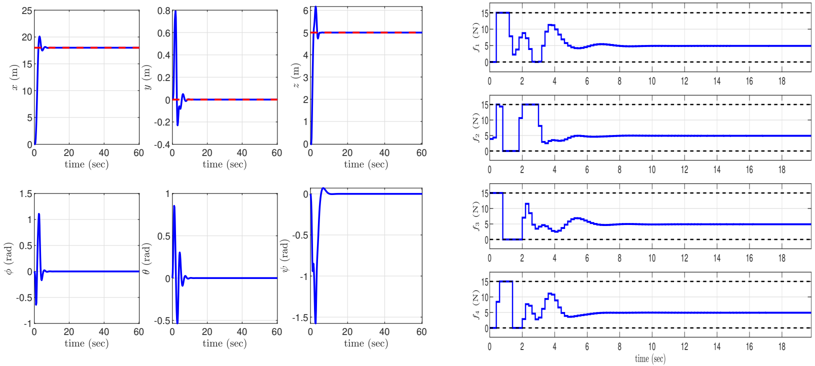
CasADi

What is CasADi?

- An open-source symbolic framework for quick, yet efficient, implementation of derivative based algorithms for dynamic optimization
- Framework for writing OCP solvers - solve NLP efficiently
- 4 standard problems can be handled by CasADi
  i.e., Quadratic Programming (QP), Nonlinear Programming (NLP), Root finding Problem, Initial-value Problem

`https://web.casadi.org/`

Ex: Position Stabilization



(a) State trajectory (b) Control input

Figure: Position stabilization of quadrotor using NMPC

Ex: Position Stabilization - cont'd

DEMO: P2P flight: simulation results

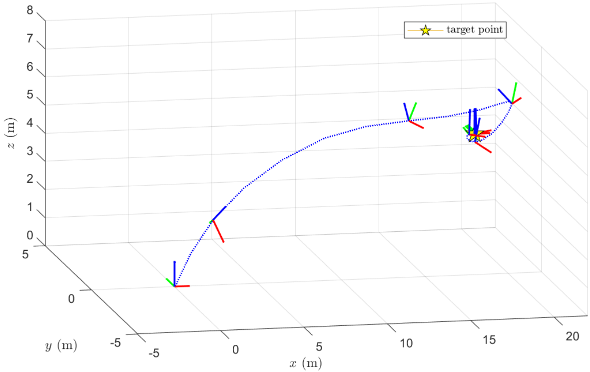
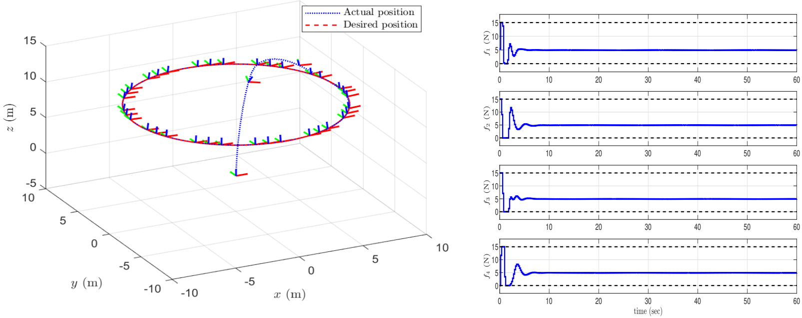


Figure: Position stabilization of quadrotor using NMPC

Ex: Circular Tracking

DEMO: simulation results

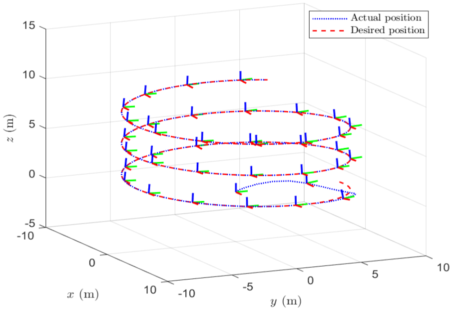


(a) 3D trajectory (b) Control input

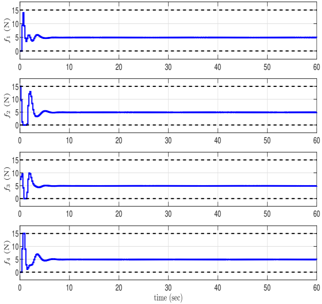
Figure: Circular tracking control of quadrotor using NMPC

Ex: Helix Tracking

DEMO: simulation results



(a) State trajectory

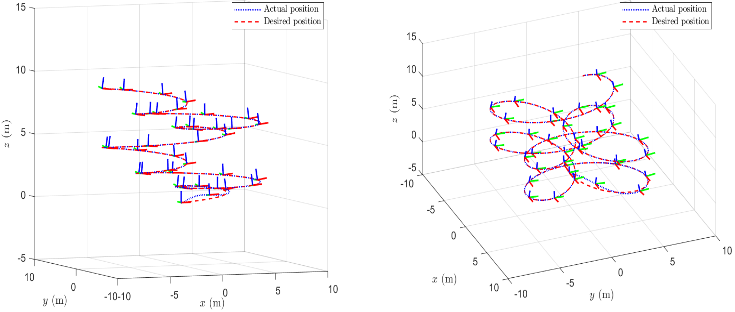


(b) Control input

Figure: Helix tracking control of quadrotor using NMPC

Ex: Complex Helix Tracking

DEMO: simulation results



(a) State trajectory (b) Control input

Figure: Complex Helix tracking control of quadrotor using NMPC

Ex: Complex Helix Tracking - cont'd

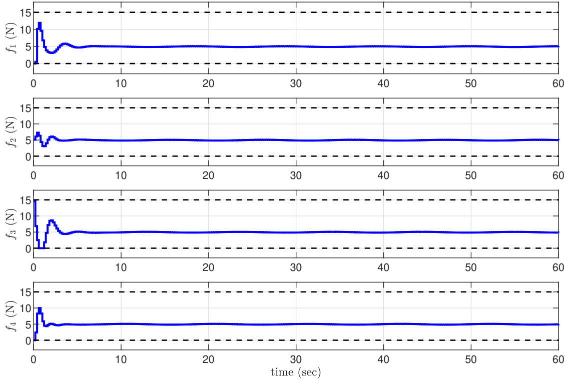


Figure: Complex Helix tracking control of quadrotor using NMPC

Outline

Conclusions

My work in this project

- Formulate quadrotor control as an optimal control problem
- Cast the OCP to NLP as a NLP via Multiple Shooting
- MPC implementation to quadrotor control using CaSAdi
  i.e., position stabilization and trajectory tracking

Future Work

- Stability Analysis, e.g. effect of length of prediction horizon, etc.

References

T. Luukkonen.
Modelling and control of quadcopter.
Independent research project in applied mathematics, Espoo (2011).

van Lookeren Campagne, Gijs.
A Nonlinear Model Predictive Control based Evasive Manoeuvre Assist Function.
TU Delft Mechanical, Maritime and Materials Engineering

References

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, vol. 24, no. 4, pp. 1394-1406, July 2016.

A. S. Hussein, C. M. Elias and E. I. Morgan
”A Realistic Model Predictive Control using Single and Multiple Shooting in the Formulation of Non-linear Programming Model”
2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Cairo, Egypt, 2019.