THE UNIVERSITY OF CALIFORNIA

LET THERE BE LIGHT

UCLA

Minimum Snap Trajectory Generation and Control for Quadrotors

Authors: Daniel Mellinger and Vijay Kumar

Preliminary Exam Paper Presentation

Zhaoliang Zheng

Electrical and Computer Engineering

The Laboratory for Embedded Machines and Ubiquitous Robots (LEMUR)

UC Los Angeles

Advisor: Dr. Ankur Mehta

Contents

Motivation

- Most of the work in this area uses controllers that are derived from linearization of the model around hover conditions and are stable only under reasonably small roll and pitch angles.
- Some work in this area has addressed aerobatic maneuvers [3, 6, 9, 10]. However, there are no stability and convergence guarantees when the attitude of the rotor craft deviates substantially from level hover conditions.
- While machine learning techniques have been successful in learning models using data from human pilots [9] and in improving performance using reinforcement learning [3], these approaches do not appear to lend themselves to motion planning or trajectory generation in environments with obstacles.
- Similar problems have been addressed using model predictive control (MPC) [11, 12]. With these approaches, guarantees of convergence are only available when the linearized model is fully controllable [12] or if a control Lyapunov function can be synthesized [13].
- As such it appears to be difficult to directly apply such techniques to the trajectory generation of a quadrotor.

Introduction

In this paper, we address the controller design and the trajectory generation for a quadrotor maneuvering in three dimensions in a tightly constrained setting typical of indoor environments. In such settings, it is necessary to develop flight plans that leverage the dynamics of the system instead of simply viewing the dynamics as a constraint on the system. It is necessary to relax small angle assumptions and allow for significant excursions from the hover state. We develop an algorithm that enables the generation of optimal trajectories through a series of keyframes or waypoints in the set of positions and orientations, while ensuring safe passage through specified corridors and satisfying constraints on achievable velocities, accelerations and inputs.

Some Notations:

- Coordinate systems: world frame $\mathcal{W}$, body frame $\mathcal{B}$
- Euler angles: roll, pitch and yaw (ϕ, θ, ψ)
- Rotation matrix from $\mathcal{B}$ to $\mathcal{W}$: ${}^{W}R_{B} = {}^{W}R_{C}\,{}^{C}R_{B}$
- Angular velocity of $\mathcal{B}$: $\omega_{\mathcal{BW}}$
- For each rotor: angular speed ω_i and force: F_i , moment M_i
- Control input: $\boldsymbol{u} = [u_1 \quad u_2 \quad u_3 \quad u_4]^T$
- COM in $\mathcal{W}$: $\boldsymbol{r} = [x \quad y \quad z]^T$
- Inertia matrix referenced to COM along $\mathcal{B}$ axes: $\mathcal{J}$
- Distance from the axis of rotation of the rotors to COM: L

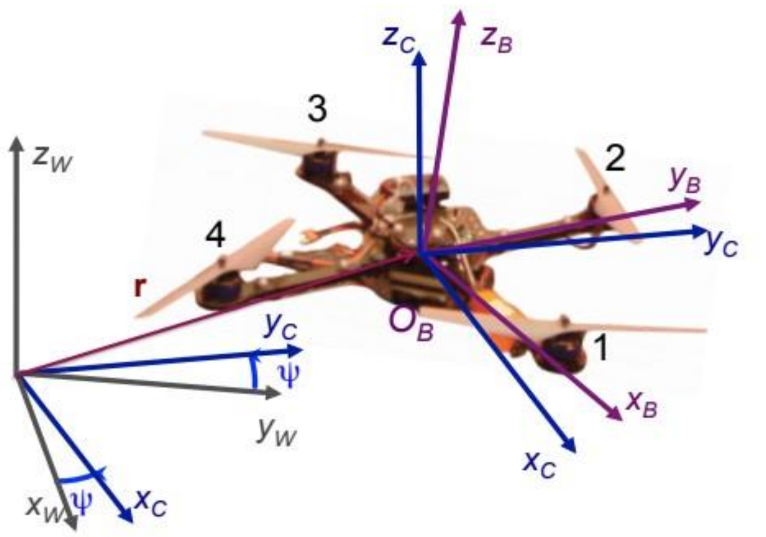


Fig. 1. The flat outputs and the reference frames.

Model

Key equations:

- $\omega_{\mathcal{BW}} = p\boldsymbol{x}_B + q\boldsymbol{y}_B + r\boldsymbol{z}_B$ (1)
- $F_i = k_F\omega_i^2\ ,\ M_i = k_M\omega_i^2$
- $\boldsymbol{u} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$ (2)
- $m\ddot{\boldsymbol{r}} = -\mathrm{mg}\mathbf{z}_\mathrm{B} + \mathrm{u}_1\mathbf{z}_\mathrm{B}$ (3)
- $\dot{\omega}_{\mathcal{BW}} = \mathcal{J}^{-1}\left[-\omega_{\mathcal{BW}} \times \mathcal{J}\omega_{\mathcal{BW}} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}\right]$ (4)
- System states:

$$\mathbf{x} = [x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad p \quad q \quad r]^T$$

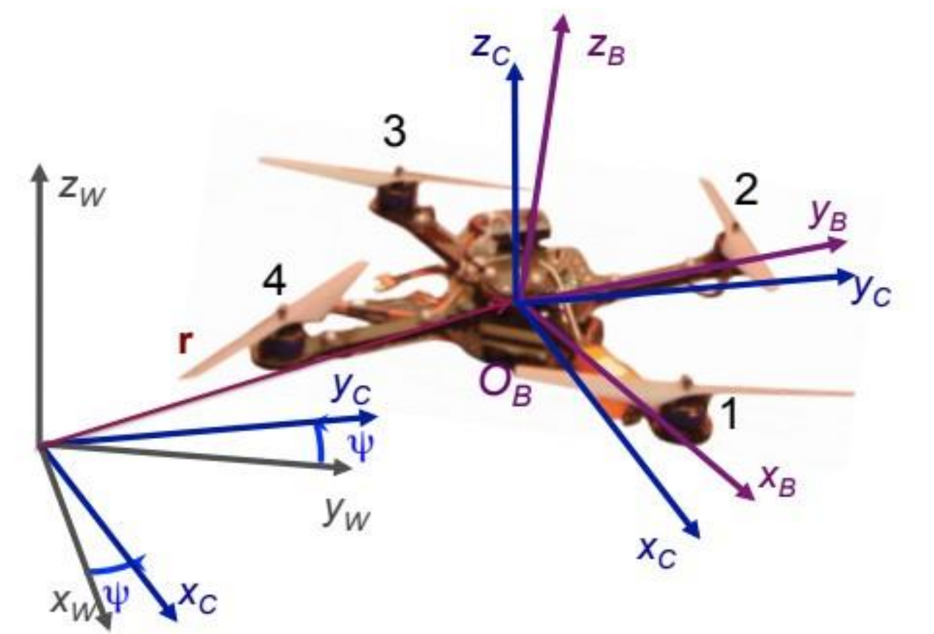


Fig. 1. The flat outputs and the reference frames.

II. Model

III. Differential Flatness

01 ? What?

02 ? ? ? Why?

03 ? ? ? -> ! ! ! Derivation!

The states and inputs are differentially flat: they can be written as algebraic functions of selected flat outputs and their derivatives:

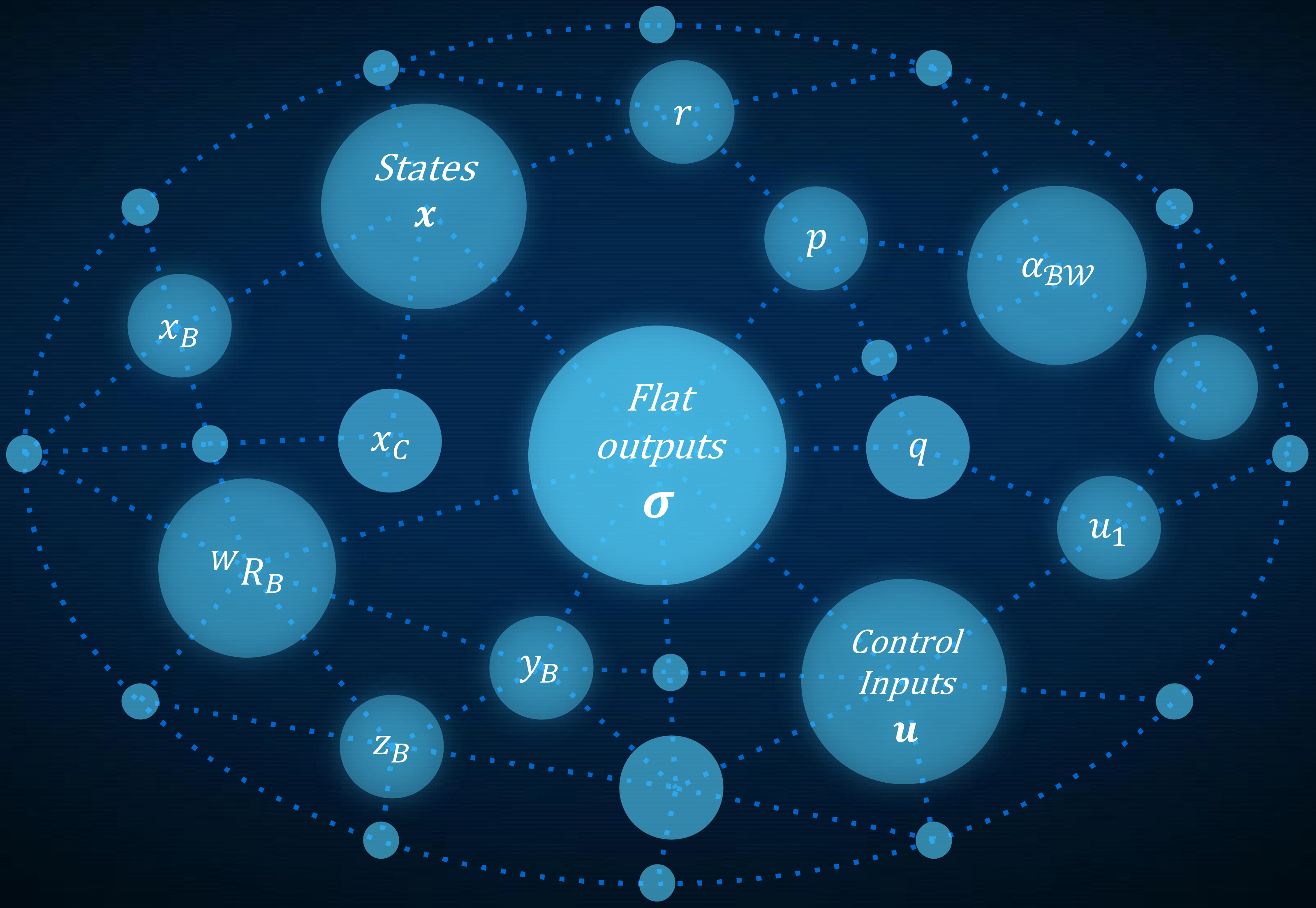
$$\mathbf{x}, \mathbf{u} = \mathbf{f}(\boldsymbol{\sigma}) = \mathbf{f}([\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\psi}]^{\boldsymbol{T}})$$

Use trajectory as input and constrains to control over all the states and control inputs:

$$\sigma(t) = [t_0, t_m] \rightarrow \mathbb{R}^3 \times SO(2)$$

Save the best to the next!

Derivation



States
$\boldsymbol{x}$
r
p
α_{BW}
x_B
x_C
Flat outputs
$\boldsymbol{\sigma}$
q
u_1
${}^W R_B$
y_B
Control Inputs
$\boldsymbol{u}$
z_B

Derivation

$\sigma(t) = [t_0, t_m] \rightarrow \mathbb{R}^3 \times SO(2)$

Position: $\sigma_{1:3} = \boldsymbol{r} = [x, y, z]^T$

Velocity: $\dot{\sigma_{1:3}} = [\dot{\boldsymbol{x}} \quad \dot{\boldsymbol{y}} \quad \dot{\boldsymbol{z}}]^T$

Acceleration: $\ddot{\sigma_{1:3}}$

$\sigma = [x, y, z, \psi]^T$

01 Trajectory, Position, velocity and acceleration of COM

02 ${}^W R_B$

$$z_B = \frac{t}{||t||}, t = [\ddot{\sigma_1}, \ddot{\sigma_2}, \ddot{\sigma_3} + g]^T$$

$$\sigma_4 = \psi \implies x_C = [\cos\sigma_4, \sin\sigma_4, 0]^T$$

$$z_B, x_C \implies \begin{cases} y_B = \dfrac{z_B \times x_C}{||z_B \times x_C||} \\ x_B = y_B \times z_B \end{cases}$$

$$\xRightarrow{x_C \times z_B \neq 0} \quad {}^W R_B = [x_B \quad y_B \quad z_B]^T$$

03 *Angular velocity:* $p\ q\ r$

First derivative of (3): $m\ddot{\boldsymbol{r}} = -\mathrm{mg}\mathbf{z_B} + \mathrm{u_1}\mathbf{z_B}$

$$\Rightarrow \quad m\dot{a} = \dot{u}_1 z_B + \omega_{BW} \times u_1 z_B \qquad (7)$$

Project this along z_B, and using the fact: $\dot{u}_1 = z_B \cdot m\dot{a}$

$$\xRightarrow{\text{substitute } \dot{u}_1 \text{ into } (7)} h_\omega = \omega_{BW} \times z_B = \frac{m}{u_1}(\dot{a} - (z_B \cdot \dot{a})z_B)$$

h_ω is the projection onto the $x_B - y_B$ *plane*:

$$\begin{cases} p = -h_\omega \cdot y_B \\ q = h_\omega \cdot x_B \end{cases}, r = \omega_{BW} \cdot z_B = (\omega_{BC} + \omega_{CW})z_B \xRightarrow{\omega_{BC} \text{ has no } z_B}$$

$$r = \omega_{CW} \cdot z_B = \dot{\psi} z_W \cdot z_B$$

04 Angular acceleration: $\boldsymbol{\alpha}_{BW}$

Control inputs: u

$\boldsymbol{\alpha}_{BW}$ along $\mathrm{x_B, y_B}$: Second derivative of (3) and follow the same process.

$\boldsymbol{\alpha}_{BW}$ along $\mathrm{z_B}$: use the fact

$$\boldsymbol{\alpha}_{BW} = \boldsymbol{\alpha}_{BC} + \omega_{CW} \times \omega_{BC} + \boldsymbol{\alpha}_{CW}$$

Note:

- $\boldsymbol{\alpha}_{BC} \cdot z_B = 0$
- $z_B \cdot \omega_{CW} \times \omega_{BC} = 0$

$$\Rightarrow \boldsymbol{\alpha}_{BW} \cdot z_B = \boldsymbol{\alpha}_{CW} \cdot z_B$$

$$= \ddot{\psi} z_W \cdot z_B$$

$$\begin{cases} u_1 = m||t|| \\ \xRightarrow{\text{Given } \boldsymbol{\alpha}_{BW}, \omega_{BW} \text{ and } (4)} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \end{cases}$$

$$\boxed{\begin{aligned} \mathbf{x} &= [x\ y\ z \quad \varphi\ \theta\ \psi \quad \dot{\boldsymbol{x}}\ \dot{\boldsymbol{y}}\ \dot{\boldsymbol{z}} \quad p\ q\ r]^T \\ u &= [u_1 \quad u_2 \quad u_3 \quad u_4]^T \end{aligned}}$$

IV. Control

The errors between specified trajectories, attitude and current trajectories, attitude:

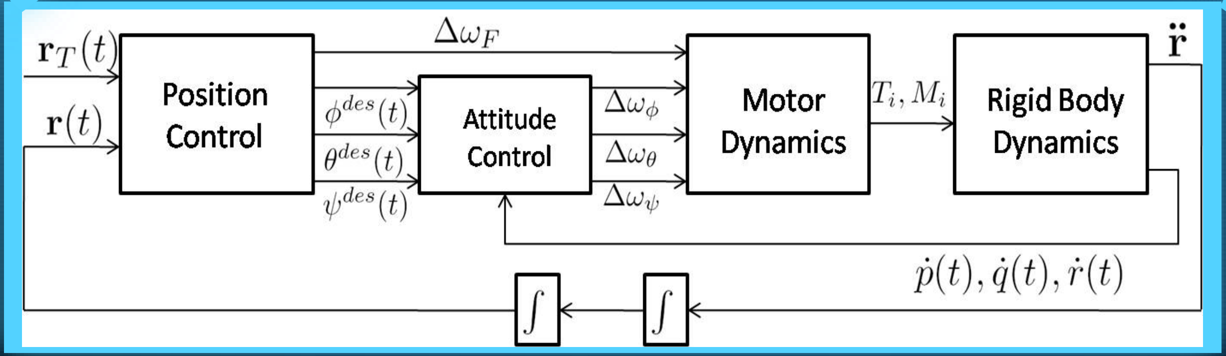
e_p, e_v, e_R, e_ω

A controller to follow specified trajectories:

$$\boldsymbol{\sigma_T(t) = [r_T(t)^T, \psi_T(t)]^T}$$

Control inputs:

$$\boldsymbol{u} = [u_1, u_2, u_3, u_4]^T$$



[15] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro uav testbed," IEEE Robotics and Automation Magazine, Sept. 2010.

IV. Control

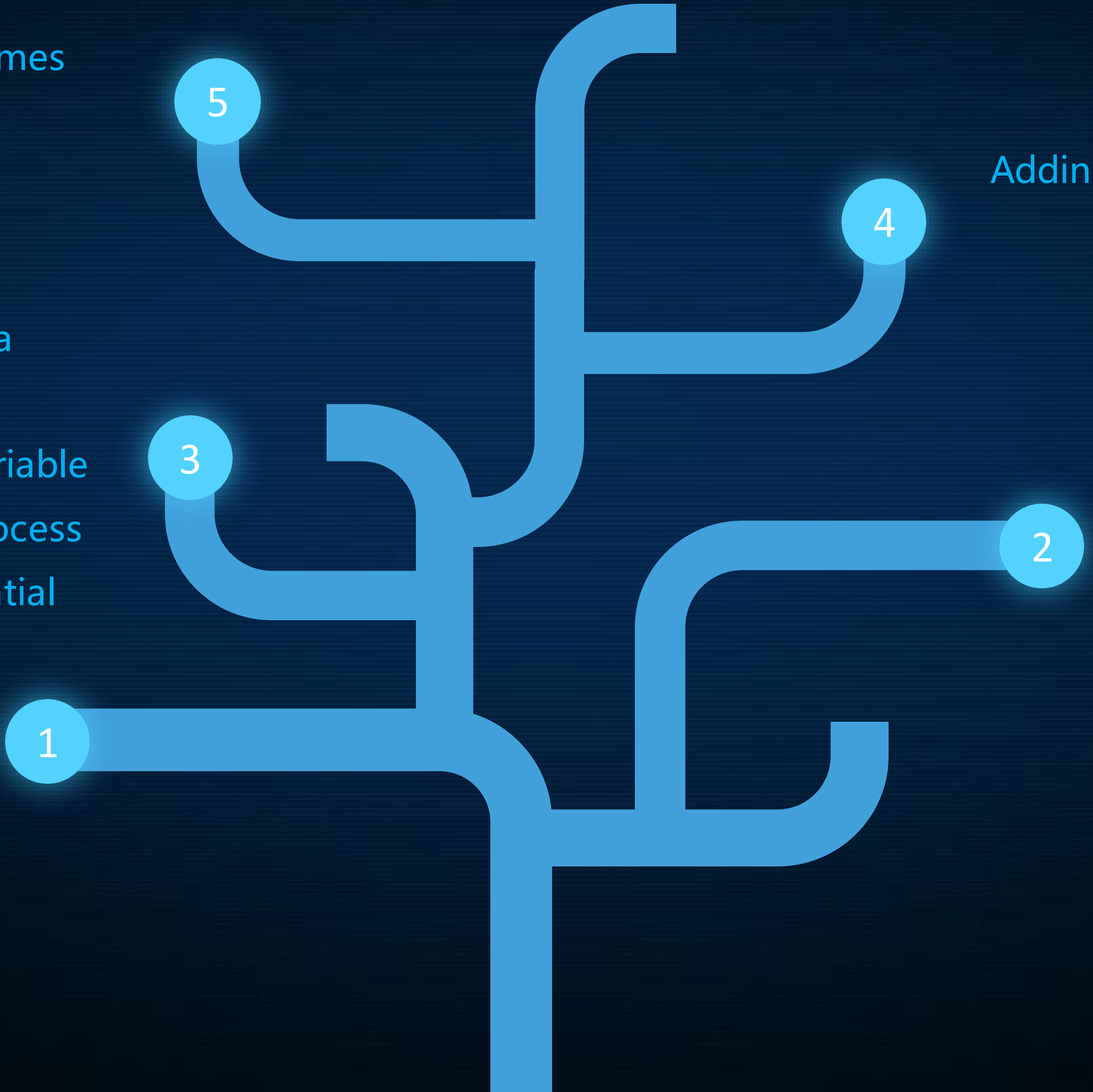
The errors between specified trajectories, attitude and current trajectories, attitude

$$
\left\{
\begin{array}{c}
\left.
\begin{array}{c}
\boldsymbol{e_p} = \boldsymbol{r} - \boldsymbol{r}_T, \boldsymbol{e}_v = \dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_T \\
\boldsymbol{F_{des}} = -\boldsymbol{K_p e_p} - \boldsymbol{K_v e_v} + mg\boldsymbol{z_W} + m\ddot{\boldsymbol{r}}_T
\end{array}
\right\} u_1 = \boldsymbol{F_{des}} \cdot z_B \\
\left.
\begin{array}{c}
\boldsymbol{z_{B,des}} = \dfrac{\boldsymbol{F_{des}}}{||\boldsymbol{F_{des}}||} \\
{}^{W}R_B \xRightarrow{\text{denote as } R_{des}} R_{des} e_3 = z_{B,des} \\
\boxed{
\begin{array}{c}
\boldsymbol{e_R} = \dfrac{1}{2}\left(\boldsymbol{R_{des}^T}\,{}^{W}R_B - {}^{W}R_B^T R_{des}\right) \\
\boldsymbol{e_\omega} = {}^{B}[\omega_{BW}] - {}^{B}[\omega_{BW,T}]
\end{array}
}
\end{array}
\right\} [\boldsymbol{u_2}, \boldsymbol{u_3}, \boldsymbol{u_4}]^T = -\boldsymbol{K_R e_R} - \boldsymbol{K_\omega e_\omega}
\end{array}
\right\}
$$

Control inputs:

$$u = [u_1, u_2, u_3, u_4]^T$$

Trajectory Generation



1. Concider trajectories in the flat output space of the form of (5).
2. Formulate the problem as a quadratic program
3. Nondimensionalization: consider a general form of the optimization problem for a nondimensional variable and nondimensional time. The process includes temporal scaling and spatial scaling.
4. Adding corridor constaints
5. Optimal segment times

VI. Experiments

Spatially Scaled Trajectories

This experiment demonstrates how the spatially scaled trajectory is used to fly through a thrown circular hoop.

Temporal Scaling, Corridor Constraints, and Optimal Segment Times

This experiment demonstrates the ability to fly through environments with several narrow gaps.

The worst case performance is for the position the farthest away (x = 1:6 meters and y = 0:4 meters) for which data is shown in Fig. 4.

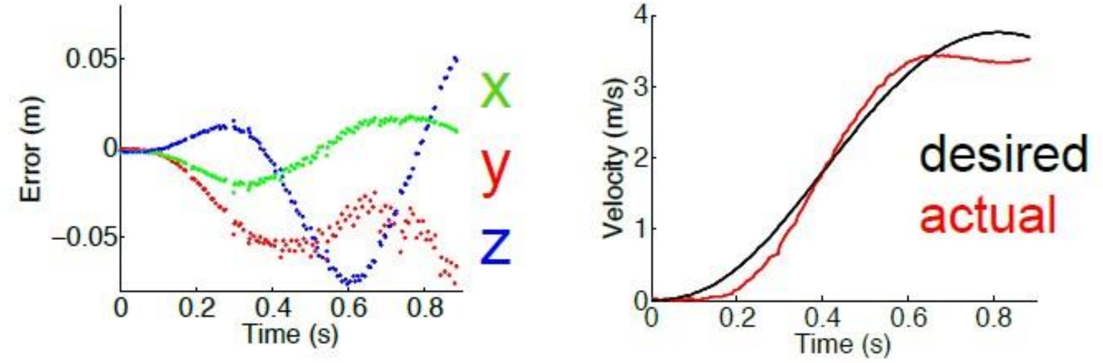


Fig. 4. Performance data for a trajectory for flying through a thrown hoop.

A series of images showing the full experiment are shown in Fig. 5.

Fig. 5. Composite image of a single quadrotor flying through a thrown circular hoop. See attached video or http://tinyurl.com/pennquad.

A. Spatially Scaled Trajectories

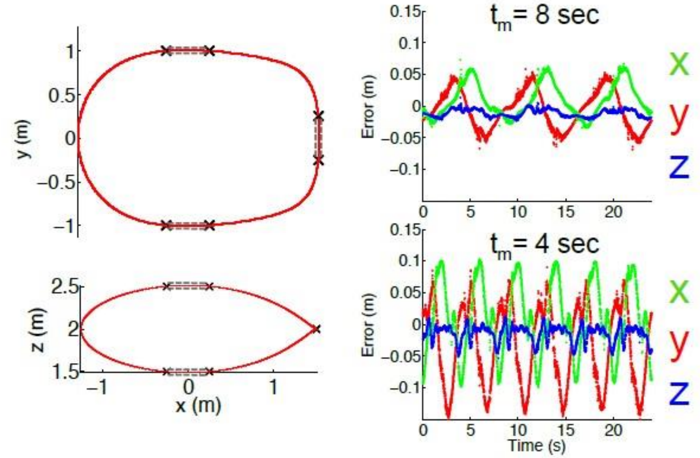


Fig. 6. Trajectory generated to fly through three gaps (left) and performance data for two traversal speeds (right).

Fig. 7. Composite image of a single quadrotor quickly flying through three static circular hoops. See attached video or http://tinyurl.com/pennquad.

B. Temporal Scaling, Corridor Constraints, and Optimal Segment Times

References

[3] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea, "A simplelearning strategy for high-speed quadrocopter multi-flips," in Proc. Of the IEEE Int. Conf. on Robotics and Automation, Anchorage, AK, May 2010, pp. 1642–1648.

Thanks for your listening

Q&A?

Back-up slides

$$\sigma_T(t) = \begin{cases} \sum_{i=0}^{n} \sigma_{Ti1} t^i & t_0 \le t \le t_1 \\ \sum_{i=0}^{n} \sigma_{Ti1} t^i & t_1 \le t \le t_2 \\ \vdots \\ \sum_{i=0}^{n} \sigma_{Tim} t^i & t_{m-1} \le t \le t_m \end{cases}$$

$$\min \int_{t_0}^{t_m} \mu_r \left\| \frac{d^{k_r} r_T}{dt^{k_r}} \right\|^2 + \mu_\psi \frac{d^{k_r} r_T}{dt^{k_r}}$$