Midterm presentation

The Quadrocopters

Technische Universität München

26. Mai 2015

Overview

1. Motivation
2. Model
3. Realtime Optimization Approach
4. Results

Optimal Control Problem

$$\min_{x,u} J(x,u) \qquad \dot{x} = f(x,u)$$

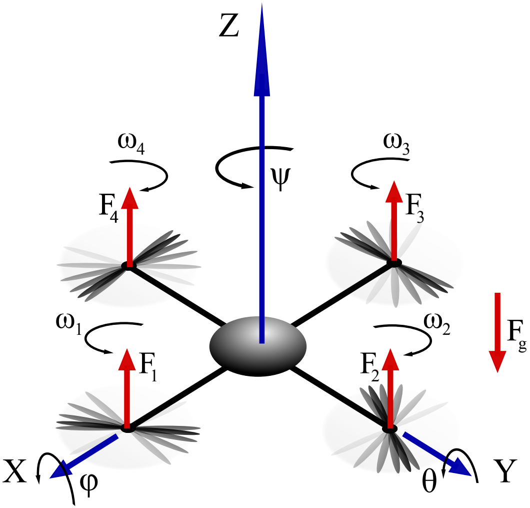
$$x: \text{ state}$$

$$u: \text{ control}$$

$\rightarrow$ additional difficulty: realtime approach

Model

Forces and Torques



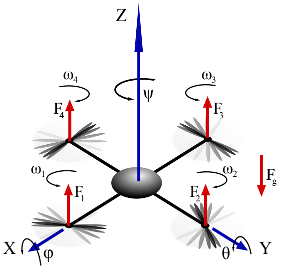
Newton-Euler Equations

Forces

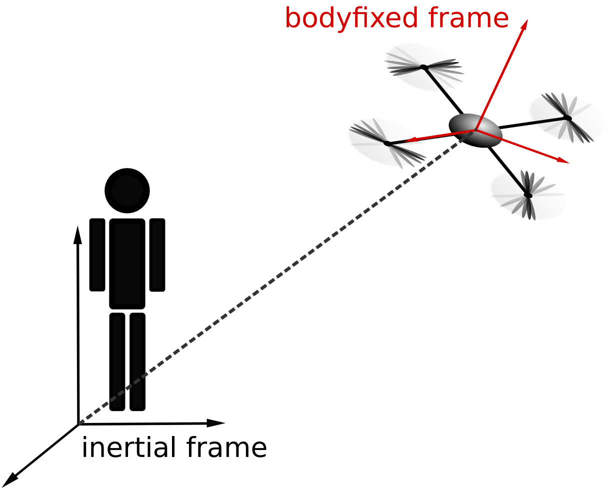
$$F_{ext} = F_g + \sum_{i=1}^{4} F_i$$

Torques

$$\tau_{ext} = \sum_{i=1}^{4} \tau_i + (\tau_\varphi + \tau_\theta)$$



Coordinate Systems



Quaternions

$$q = a + \mathrm{i}b + \mathrm{j}c + \mathrm{k}d \qquad a, b, c, d \in \mathbb{R}$$

Quaternions

$$q = a + \mathrm{i}b + \mathrm{j}c + \mathrm{k}d \qquad a, b, c, d \in \mathbb{R}$$

representing rotation $\Leftrightarrow \|q\| = 1$

Quaternions

$$q = a + \mathrm{i}b + \mathrm{j}c + \mathrm{k}d \qquad a, b, c, d \in \mathbb{R}$$

representing rotation $\Leftrightarrow \|q\| = 1$

advantage $\rightarrow$ no singularities

Quaternions

$$q = a + \mathrm{i}b + \mathrm{j}c + \mathrm{k}d \qquad a, b, c, d \in \mathbb{R}$$

representing rotation $\Leftrightarrow \|q\| = 1$

advantage $\rightarrow$ no singularities

problem $\rightarrow \|q\| = 1$ additional constraint

Dynamics

Equations representing dynamics...

$$T(x, u) = M \cdot \begin{pmatrix} \dot{x}_8 \\ \vdots \\ \dot{x}_{13} \end{pmatrix} + \Theta(x)$$

Dynamics

Equations representing dynamics...

$$T(x,u) = M \cdot \begin{pmatrix} \dot{x}_8 \\ \vdots \\ \dot{x}_{13} \end{pmatrix} + \Theta(x)$$

...expressed as system of differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} x_1 \\ \vdots \\ x_7 \\ x_8 \\ \vdots \\ x_{13} \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_7 \\ M^{-1}(T(x,u) - \Theta(x)) \end{pmatrix}$$

Prospect

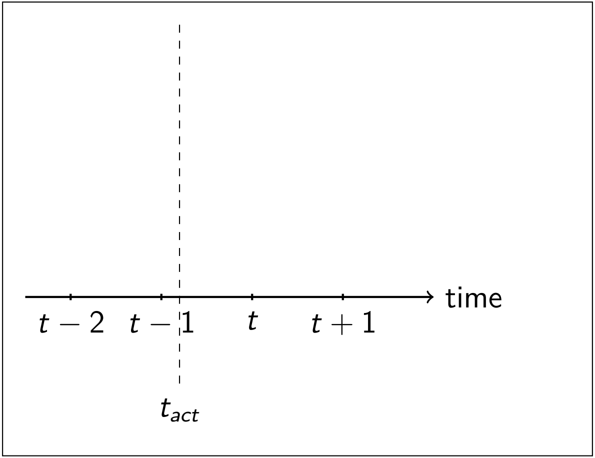
Refinement of the model

$\rightarrow$ wind

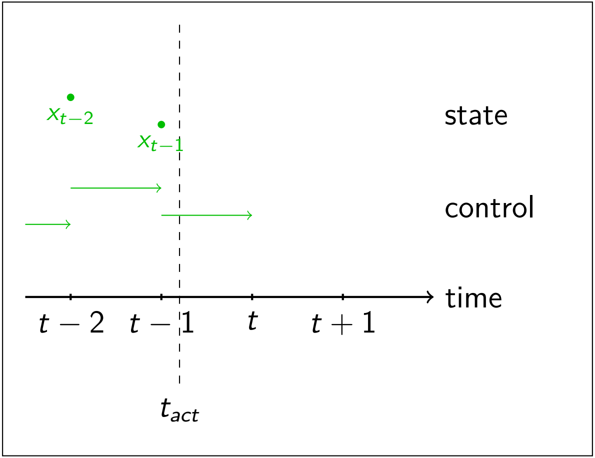
$\rightarrow$ aerodynamical forces

Realtime Optimization Approach

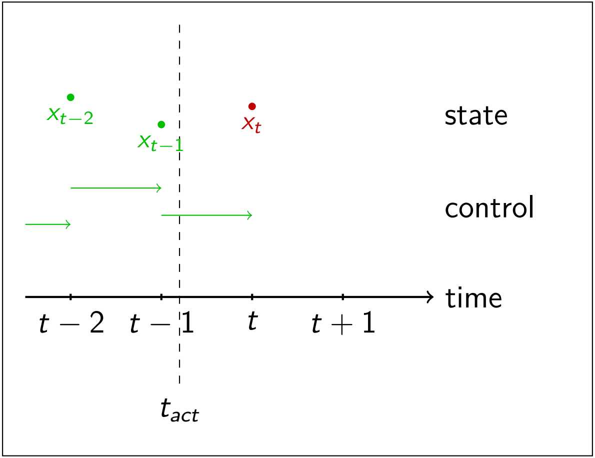
Setting



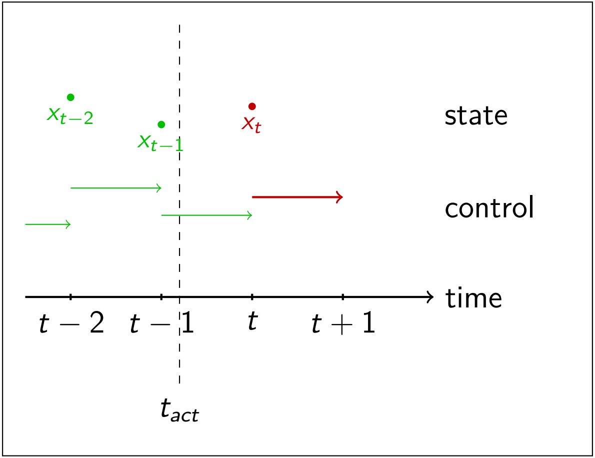
Setting



Setting



Setting



Minimization Problem

$$\min_{\substack{s_t, \ldots, s_N \\ q_t, \ldots, q_{N-1}}} \sum_{i=t}^{N-1} J_i(s_i, q_i) \quad s.t. \begin{cases} x_t - s_t = 0 \\ h_i(s_i, q_i) - s_{i+1} = 0 \quad \forall i = t, \ldots, N-1 \end{cases}$$

$J_i(s_i, q_i)$	discretized goal function
$x_t - s_t = 0$	expected state = real state
$h_i(s_i, q_i)$	solution of the ODE at time i

The Lagrangian

$$L^t(y) = \sum_{i=t}^{N-1} J_i(s_i, q_i) + \lambda_t^T(x_t - s_t) + \sum_{i=t}^{N-1} \lambda_{i+1}^T(h_i(s_i, q_i) - s_{i+1})$$

$$y := (\lambda_t, s_t, q_t, \lambda_{t+1}, s_{t+1}, q_{t+1}, ..., \lambda_N, s_N)$$

We are looking for y^* satisfying the KKT conditions:

$$\Rightarrow \nabla_y L^t(y^*) = 0$$

The SQP Method

How to find y^*?

$$y_{k+1} = y_k + \alpha_k \Delta y_k$$

$$\min_{\Delta y_k} = \frac{1}{2} \Delta y_k^T A_k \Delta y_k + \nabla_{y_k} J(y_k)^T \Delta y_k$$

The SQP Method

How to find y^*?

$$y_{k+1} = y_k + \alpha_k \Delta y_k$$
$$\min_{\Delta y_k} = \frac{1}{2} \Delta y_k^T A_k \Delta y_k + \nabla_{y_k} J(y_k)^T \Delta y_k$$

Special case:

$A_k = H(y_k)$ approximated Hessian $\nabla^2_{y_k} L(y_k)$

$\alpha_k = 1$

Newton-Raphson

Find Δy_k with:

$$\nabla_{y_k} L(y_k) + H(y_k)\Delta y_k = 0$$

Newton-Raphson

Find Δy_k with:

$$\nabla_{y_k} L(y_k) + H(y_k)\Delta y_k = 0$$

There is only time for 1 SQP step:

$$y_{t+1} = y_1 = y_0 + \Delta y_0$$

Riccati Recursion

Approximated Hessian:

$$
H^t(y^t)=\begin{pmatrix}
 & -E & & & & & & & & & \\
-E & Q_t^H & M_t^H & A_t^T & & & & & & & \\
 & (M_t^T)^H & R_t^H & B_t^T & & & & & & & \\
 & A_t & B_t & & -E & & & & & & \\
 & & & -E & Q_{t+1}^H & M_{t+1}^H & A_{t+1}^T & & & & \\
 & & & & (M_{t+1}^T)^H & R_{t+1}^H & B_{t+1}^T & & & & \\
 & & & & A_{t+1} & B_{t+1} & & & & & \\
 & & & & & & & \ddots & & & \\
 & & & & & & \ddots & Q_{N-1}^H & M_{N-1}^H & A_{N-1}^T & \\
 & & & & & & & (M_{N-1}^T)^H & R_{N-1}^H & B_{N-1}^T & \\
 & & & & & & & A_{N-1} & B_{N-1} & & -E \\
 & & & & & & & & & -E & Q_N^H
\end{pmatrix}
$$

Summary

What happens in interval $[t_{k-1}, t_k]$?

Summary

What happens in interval $[t_{k-1}, t_k]$?

1. calculate control u_{k-1} (Riccati Part II)

Summary

What happens in interval $[t_{k-1}, t_k]$?

1. calculate control u_{k-1} (Riccati Part II)
2. calculate y_k (Riccati Part II)

Summary

What happens in interval $[t_{k-1}, t_k]$?

1. calculate control u_{k-1} (Riccati Part II)
2. calculate y_k (Riccati Part II)
3. prepare u_k (Newton & Riccati Part I)

Finite Horizon

$$\min_{\substack{s_t,\dots,s_N \\ q_t,\dots,q_{N-1}}} \sum_{i=t}^{N-1} J_i(s_i, q_i) \quad s.t. \left\{ \begin{array}{c} x_t - s_t = 0 \\ h_i(s_i, q_i) - s_{i+1} = 0 \quad \forall i = t, \dots, N-1 \end{array} \right.$$

How to choose N?

$N = t_{end}$ $\quad\rightarrow$ problem size decreasing

Finite Horizon

$$\min_{\substack{s_t, \ldots, s_N \\ q_t, \ldots, q_{N-1}}} \sum_{i=t}^{N-1} J_i(s_i, q_i) \quad s.t. \left\{ \begin{array}{c} x_t - s_t = 0 \\ h_i(s_i, q_i) - s_{i+1} = 0 \quad \forall i = t, \ldots, N-1 \end{array} \right.$$

How to choose N?

$N = t_{end} \quad \rightarrow$ problem size decreasing
$N = t + n \quad \rightarrow$ problem size constant

Results

Questions?

S. Boyd.
Solving the lqr problem by block elimination.
Lecture notes, 2009.

J. Diebel.
Representing attitude: Euler angles, unit quaternions, and rotation vectors.
10 2006.

M. Diehl, B. H. Georg, J. P. Schlder, R. Findeisen, Z. Nagy, and F. Allgwer.
Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations.
Journal of Process Control, 2002.

M. Diehl, H. G. Bock, and J. P. Schlder.
A real-time iteration scheme for nonlinear optimization in optimal feedback control.
SIAM J. Control Optim., 2005.

M. M. Diehl.
Real-Time Optimization for Large Scale Nonlinear Processes.
PhD thesis, Ruprecht-Karls-Universiteidelberg, 2001.

L. R. Garcia Carrillo, A. E. Dzul Lopez, R. Lozano, and C. Pegard.
Quad Rotorcraft Control.
Springer-Verlag London, 2013.

D. Hartmann, K. Landis, M. Mehrer, S. Moreno, and J. Kim.
Quadcopter dynamic modeling and simulation (quad-sim) v1.00.
Git, 2014.
URL `https://github.com/dch33/Quad-Sim`.

E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard.
Lqr control for a quadrotor using unit quaternions: Modeling and simulation.
IEEE Xplore, 2013.

J. Richter-Gebert and T. Orendt.
Geometriekalkle.
Springer: Berlin, Heidelberg, 2009.