

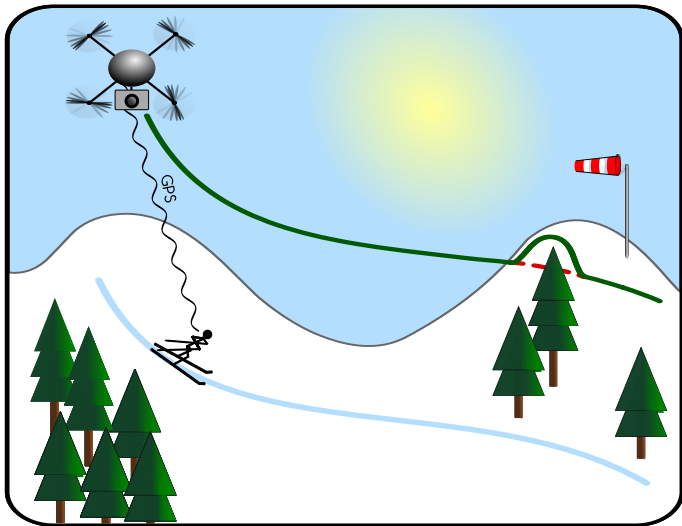
Real Time Control of a Quadcopter

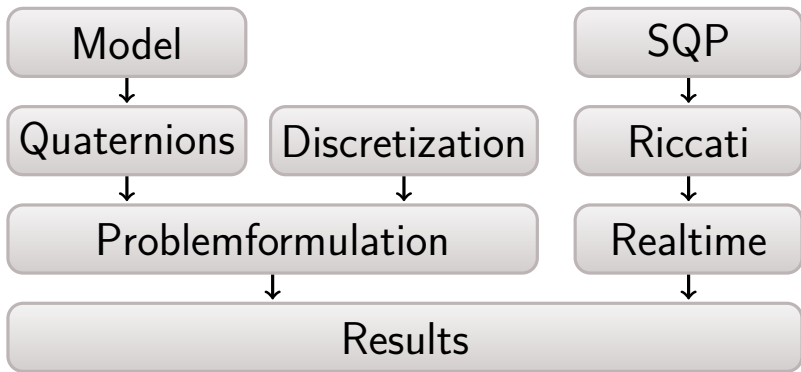
Simon Kick, Philipp Fröhlich, Benedikt König, Annika Stegie

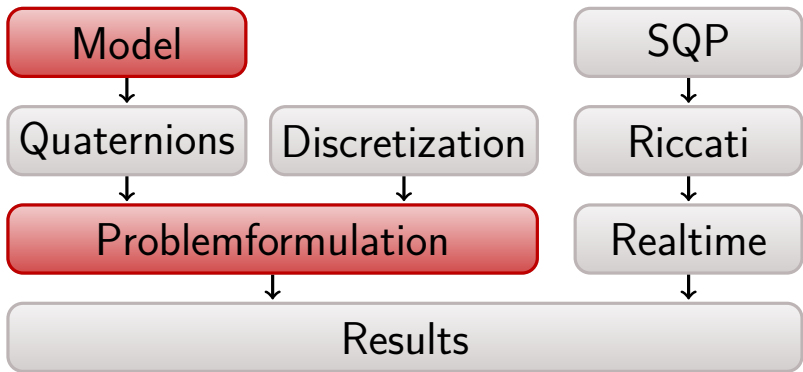
Technische Universität München

11 July 2015

Motivation







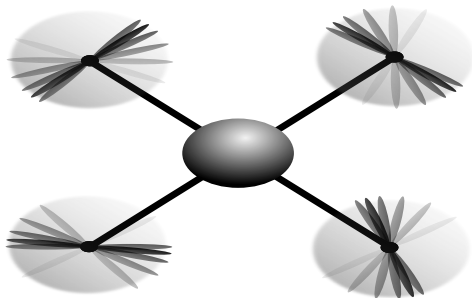
Optimal Control Formulation

$$\min_{x,u} J(x,u) \quad \text{s.t.} \quad \begin{aligned} \tilde{h}(x,u) &= 0 \\ \dot{x}(t) &= f(x(t), u(t)) \end{aligned}$$

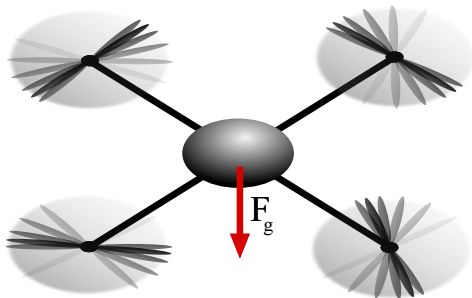
x : state

u : control

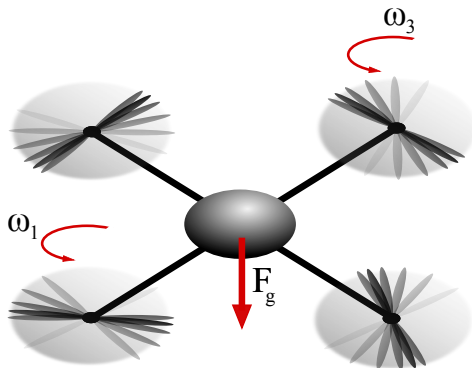
Model



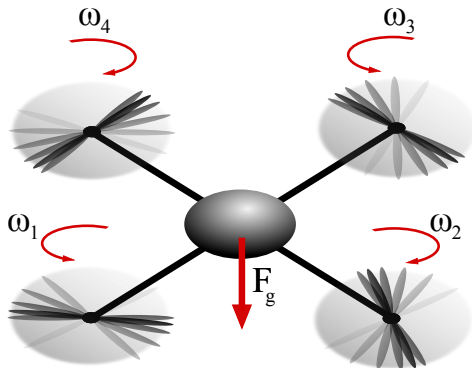
Model



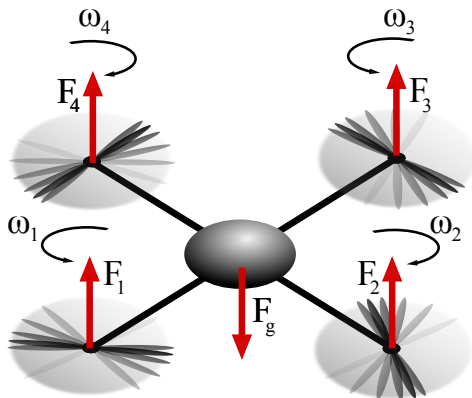
Model



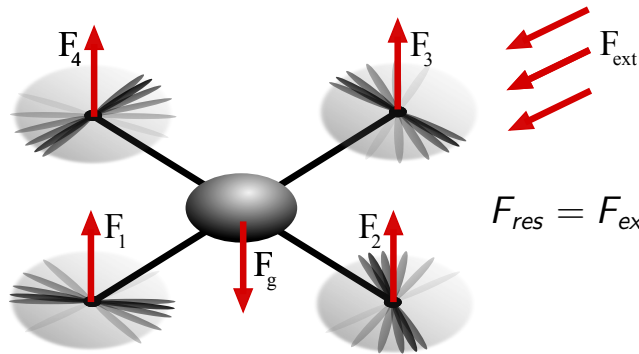
Model



Model

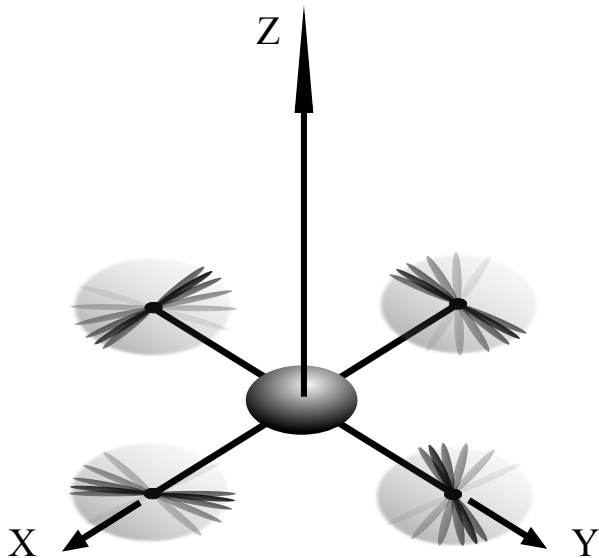


Forces

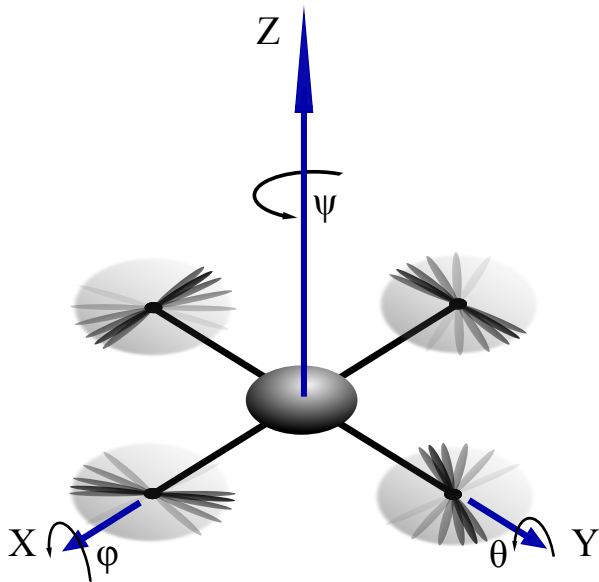


$$F_{res} = F_{ext} + F_g + \sum_{i=1}^4 F_i$$

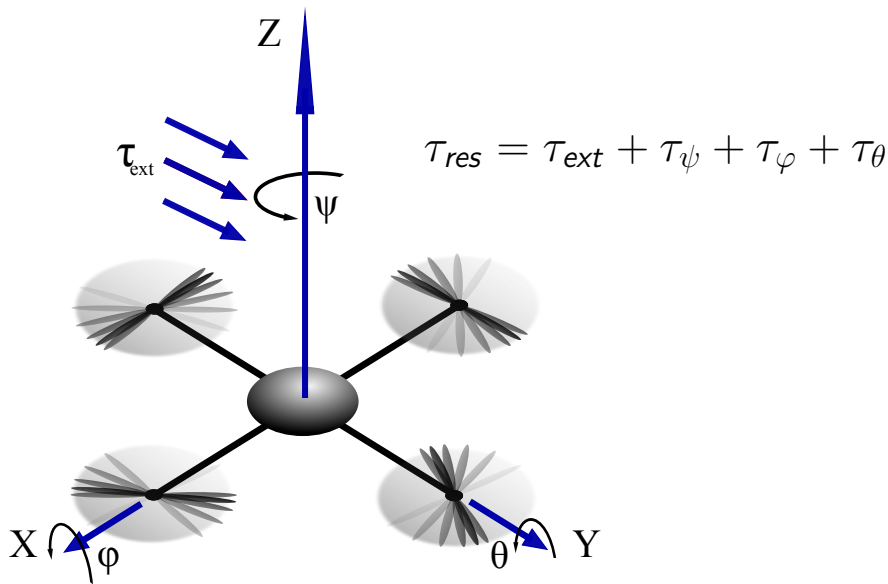
Torques



Torques



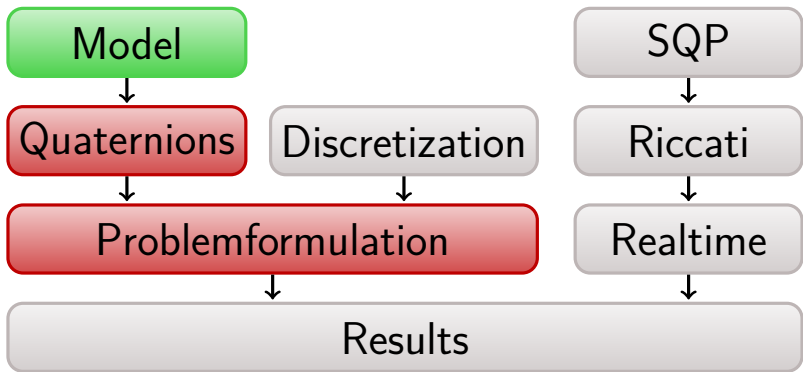
Torques



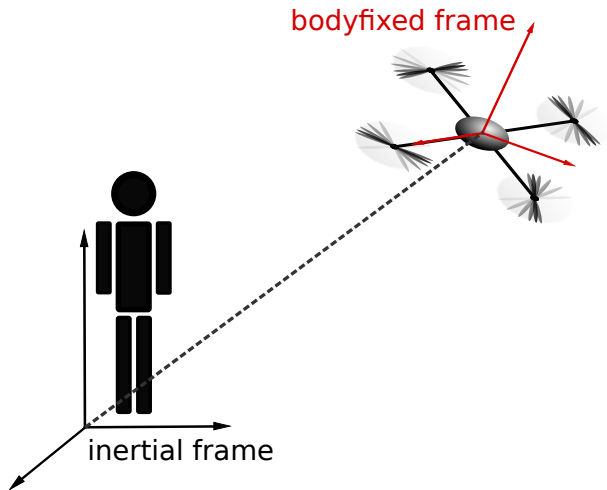
Obtain ODE

$$\left. \begin{aligned} F_{res} &= F_{ext} + F_g + \sum_{i=1}^4 F_i \\ \tau_{res} &= \tau_{ext} + \tau_{\psi} + \tau_{\varphi} + \tau_{\theta} \end{aligned} \right\} \Rightarrow \dot{x}(t) = f(x(t), u(t))$$

$$\left. \begin{aligned} \tilde{h}(x, u) &= 0 \\ \dot{x}(t) &= f(x(t), u(t)) \end{aligned} \right\} \Rightarrow h(x, u) = 0$$



Copter Rotation



Quaternions

$$q = a + ib + jc + kd \quad a, b, c, d \in \mathbb{R}$$

$$\Leftrightarrow$$

$$q = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \in \mathbb{R}^4$$

Quaternions

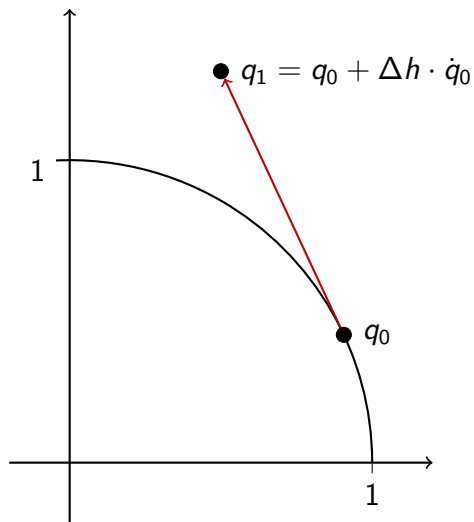
$$q = a + ib + jc + kd \quad a, b, c, d \in \mathbb{R}$$

$$\Leftrightarrow$$

$$q = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \in \mathbb{R}^4$$

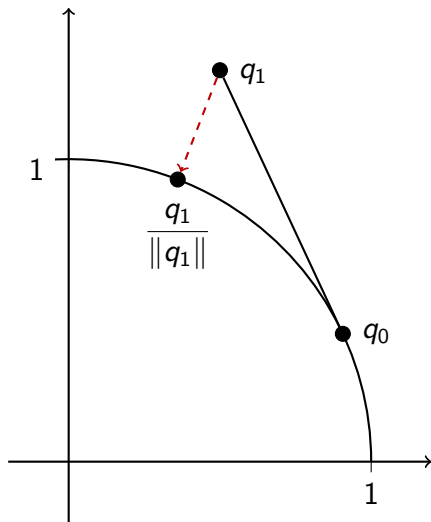
$$\text{represent rotation} \quad \Leftrightarrow \quad \|q\| = 1 \quad \Leftrightarrow \quad q \in \mathcal{S}^3$$

Drift Correction

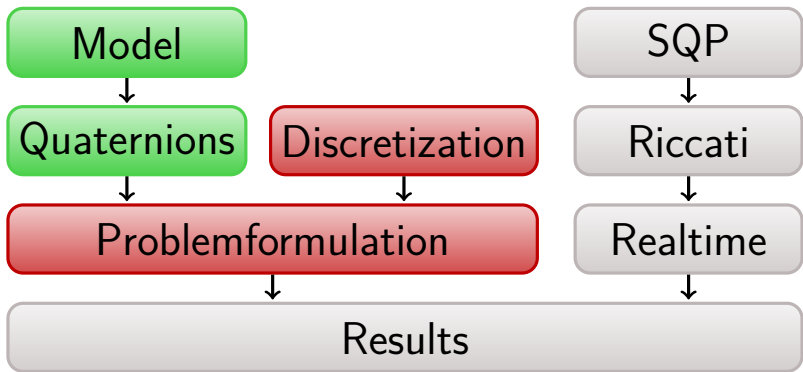


$$\dot{q}(t) = \tilde{f}(q(t))$$

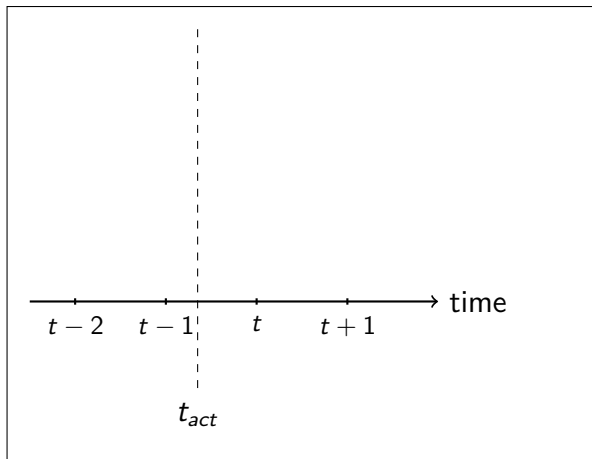
Drift Correction



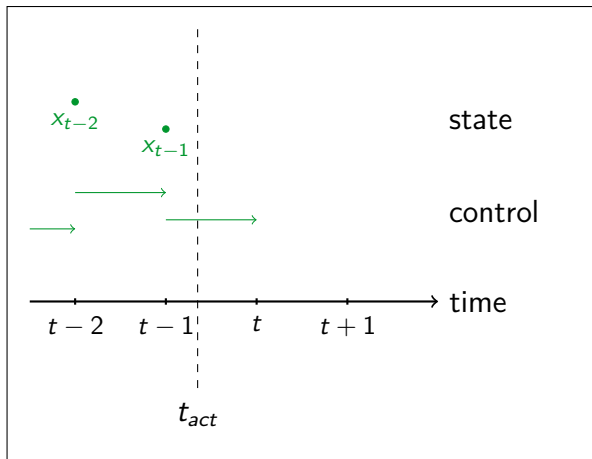
$$\dot{q}(t) = \tilde{f}(q(t)) - \lambda(q(t))$$



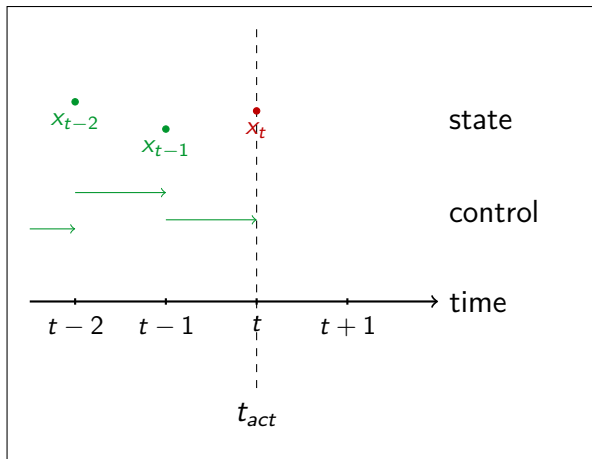
Setting



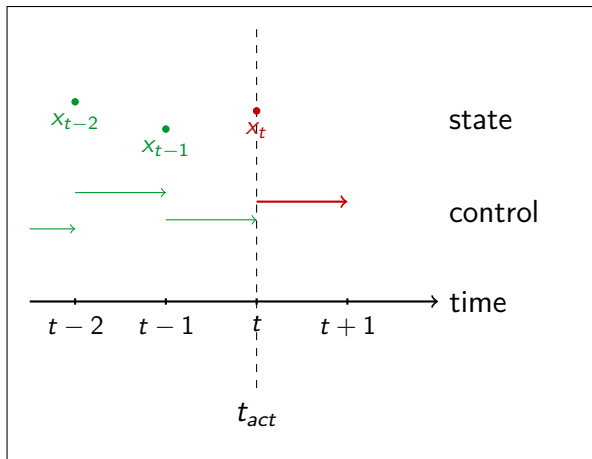
Setting



Setting



Setting

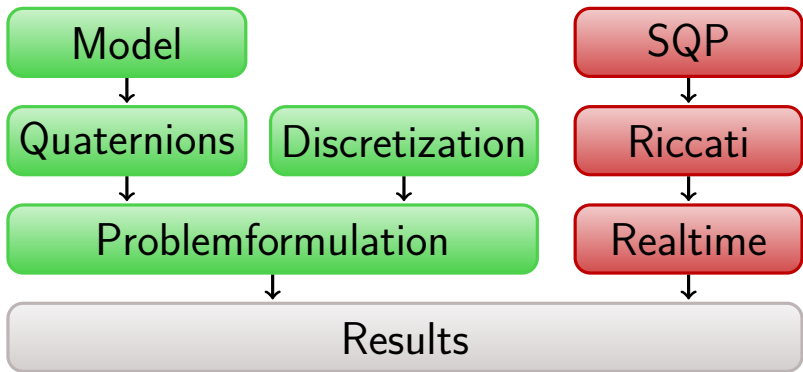


Discrete Problem

$$\min_{x,u} \sum_{i=t}^{t+N} J_i(x_i, u_i) \quad \text{s.t.} \quad h_i(x_i, u_i) = 0 \quad i = t, \dots, t + N$$

$J_i(x_i, u_i)$ discretized goal function

$h_i(x_i, u_i)$ equality constraints at time i



The Lagrangian

$$L(y) = \sum_{i=t}^{t+N} J_i(x_i, u_i) + \sum_{i=t}^{t+N} \lambda_i^T h_i(x_i, u_i)$$

$$y := (\lambda, x, u)$$

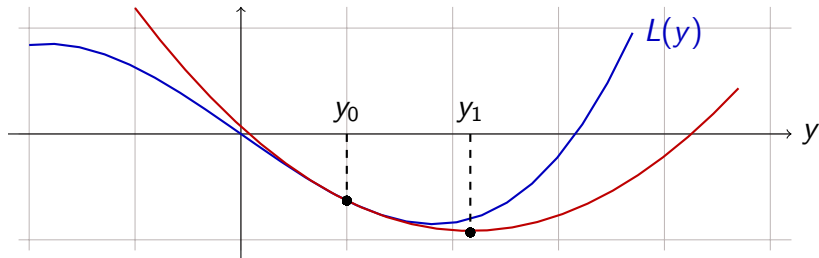
$$y^* \text{ optimal} \quad \Leftrightarrow \quad \nabla_y L(y^*) = 0$$

The SQP Method

Find y^* :

$$y_1 = y_0 + s$$

$$\min_s \frac{1}{2} s^T \nabla^2 L(y_0) s + \nabla L(y_0)^T s$$



Quasi Newton-Method

Find s with:

$$\nabla L(y_0) + \nabla^2 L(y_0)s = 0$$

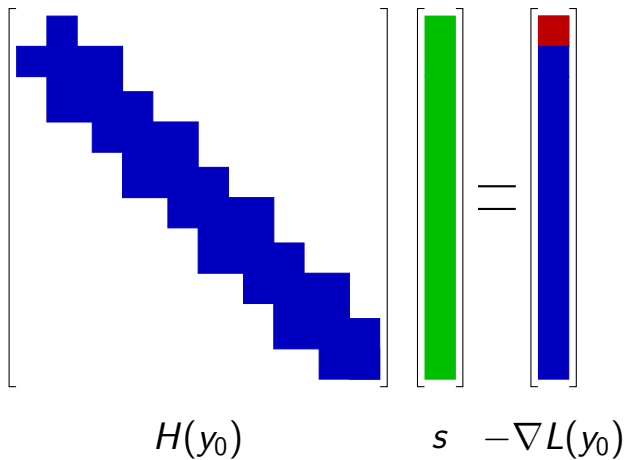
Approximate $\nabla^2 L(y_0)$ and solve:

$$H(y_0)s = -\nabla L(y_0)$$

Riccati Recursion

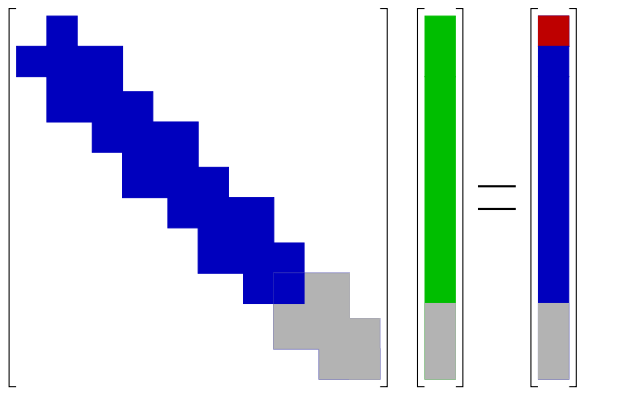
$$H(y_0) s = -\nabla L(y_0)$$

Riccati Recursion


$$\begin{bmatrix} \text{Blue Block Matrix} \end{bmatrix} = \begin{bmatrix} \text{Green Vector} \end{bmatrix} = \begin{bmatrix} \text{Red Top, Blue Bottom Vector} \end{bmatrix}$$

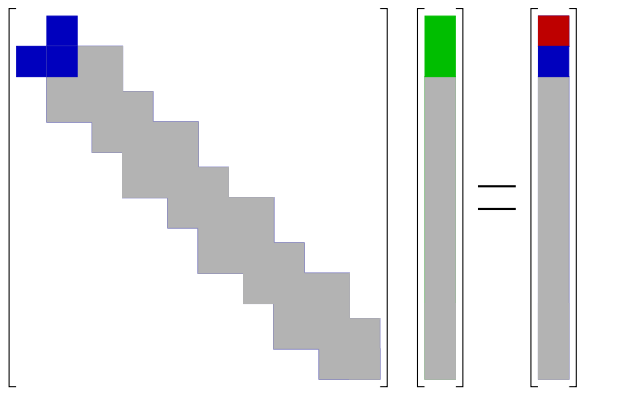
$H(y_0) \qquad s \quad -\nabla L(y_0)$

Riccati Recursion


$$H(y_0) = \begin{bmatrix} s \\ -\nabla L(y_0) \end{bmatrix}$$

The diagram illustrates the Riccati Recursion equation. It shows a large blue block matrix $H(y_0)$ on the left, followed by an equals sign, and then two vertical vectors on the right. The first vector is labeled s and consists of a green top part and a gray bottom part. The second vector is labeled $-\nabla L(y_0)$ and consists of a red top part, a blue middle part, and a gray bottom part. The blue part of the second vector is the same size as the blue part of the matrix $H(y_0)$.

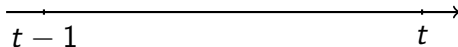
Riccati Recursion


$$\begin{bmatrix} \text{blue cross} & & & \\ & \text{gray staircase} & & \\ & & \ddots & \\ & & & \text{gray block} \end{bmatrix} = \begin{bmatrix} \text{green block} \\ \text{gray block} \end{bmatrix} = \begin{bmatrix} \text{red block} \\ \text{blue block} \\ \text{gray block} \end{bmatrix}$$

$H(y_0) \qquad s \quad -\nabla L(y_0)$

Summary

What happens in interval $[t - 1, t]$?



Summary

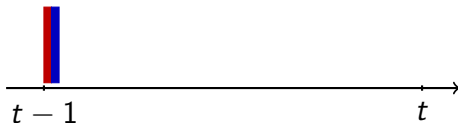
What happens in interval $[t - 1, t]$?



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

Summary

What happens in interval $[t - 1, t]$?



- 1 calculate control u_{t-1} (Riccati Part II)
- 2 calculate y (Riccati Part II)

Summary

What happens in interval $[t - 1, t]$?



- 1 calculate control u_{t-1} (Riccati Part II)
- 2 calculate y (Riccati Part II)
- 3 prepare u_t (Newton & Riccati Part I)

Summary

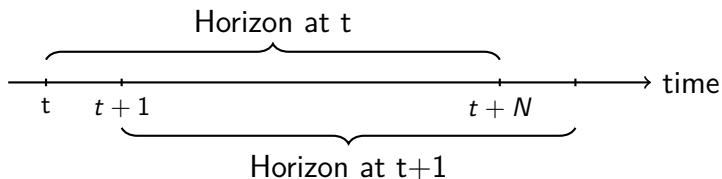
What happens in interval $[t - 1, t]$?



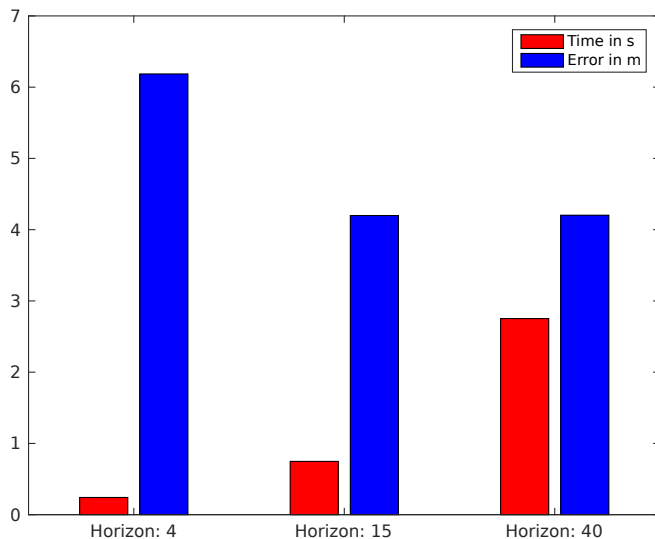
- 1 calculate control u_{t-1} (Riccati Part II)
- 2 calculate y (Riccati Part II)
- 3 prepare u_t (Newton & Riccati Part I)

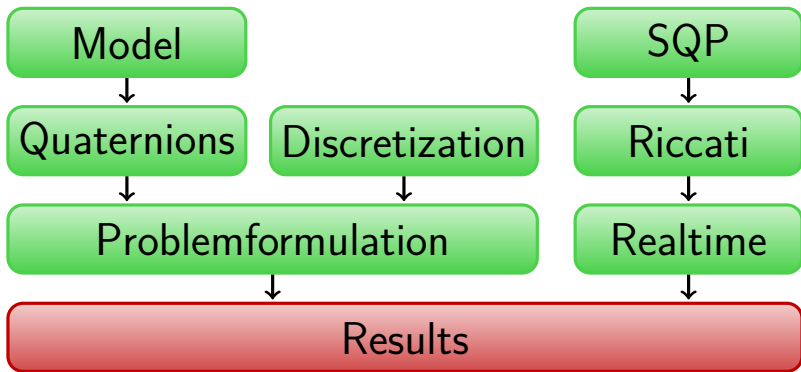
\Rightarrow with horizon 18 this is 28% faster.

Finite Horizon

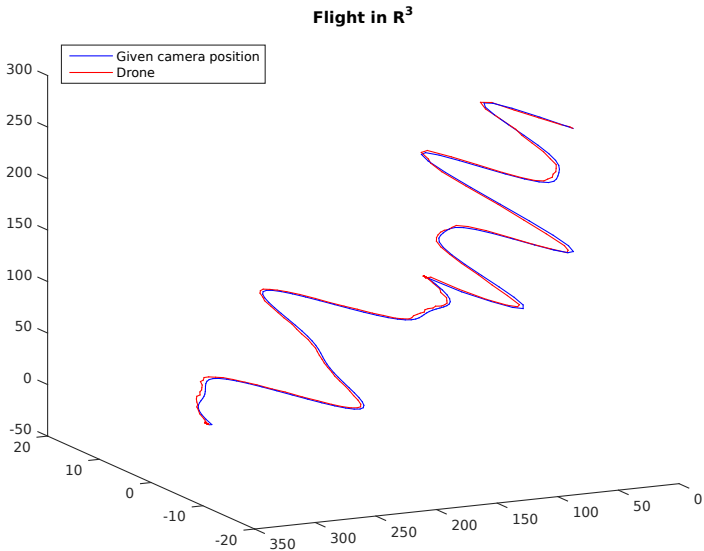


Finite Horizon

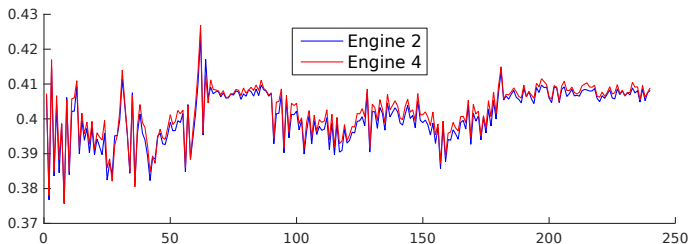
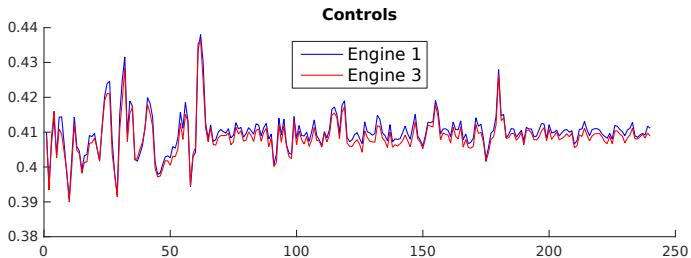




Following a Skier



Following a Skier



References I



Stephen Boyd.

Solving the lqr problem by block elimination.

Lecture notes, 2009.



James Diebel.

Representing attitude: Euler angles, unit quaternions, and rotation vectors.

10 2006.



Moritz Diehl, Hans Georg Bock, and Johannes P. Schlöder.

A real-time iteration scheme for nonlinear optimization in optimal feedback control.

SIAM J. Control Optim., 2005.

References II



Moritz Diehl, Bock H. Georg, Johannes P. Schlöder, Rolf Findeisen, Zoltan Nagy, and Frank Allgöwer.

Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations.

Journal of Process Control, 2002.



Moritz Mathias Diehl.

Real-Time Optimization for Large Scale Nonlinear Processes.

PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2001.



Luis Rodolfo Garcia Carrillo, Alejandro Enrique Dzul Lopez, Rogelio Lozano, and Claude Pegard.

Quad Rotorcraft Control.

Springer-Verlag London, 2013.

References III



D. Hartmann, K. Landis, M. Mehrer, S. Moreno, and J. Kim.
Quadcopter dynamic modeling and simulation (quad-sim) v1.00.
Git, 2014.



Elias Reyes-Valeria, Rogerio Enriquez-Caldera, Sergio
Camacho-Lara, and Jose Guichard.
Lqr control for a quadrotor using unit quaternions: Modeling and
simulation.
IEEE Xplore, 2013.



Jürgen Richter-Gebert and Thorsten Orendt.
Geometriealküle.
Springer: Berlin, Heidelberg, 2009.

What we have learned from the Project:

-
- MATLAB[®] is special
- write tests!!!
- ...