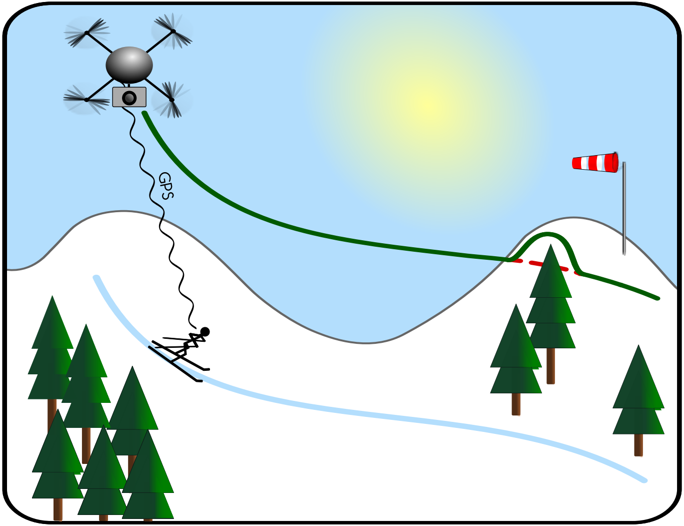
Real Time Control of a Quadcopter

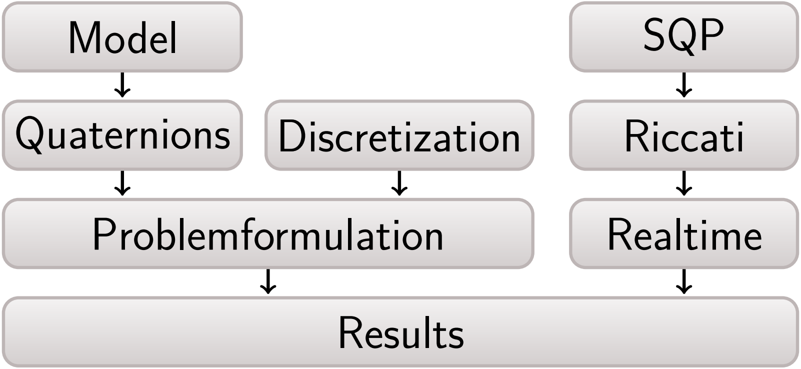
Simon Kick, Philipp Fröhlich, Benedikt König, Annika Stegie

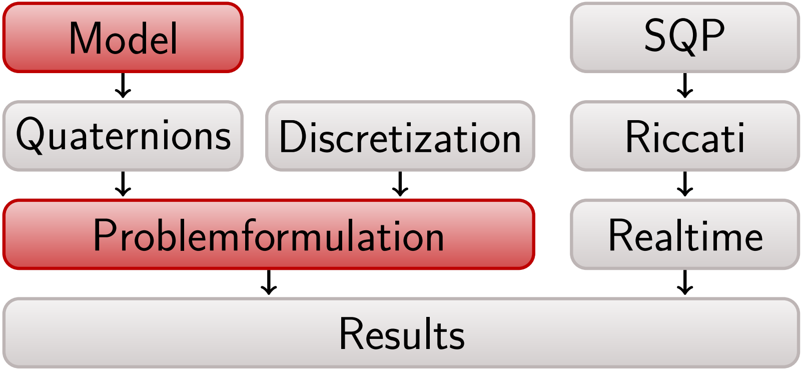
Technische Universität München

11 July 2015

Motivation




Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results


Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results

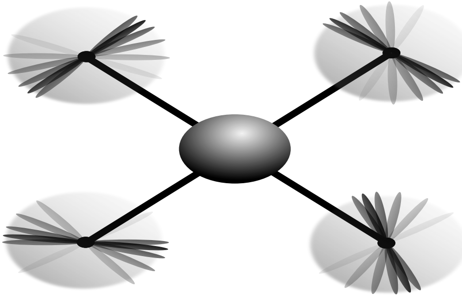
Optimal Control Formulation

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

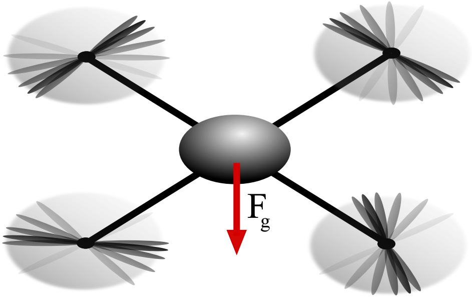
x : state

u : control

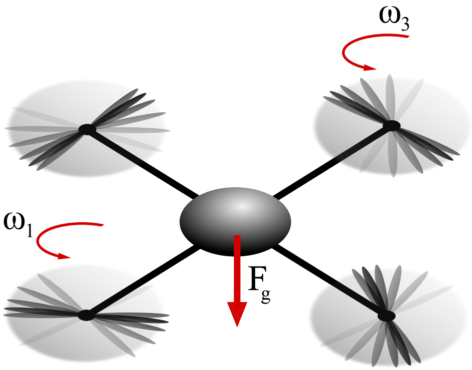
Model



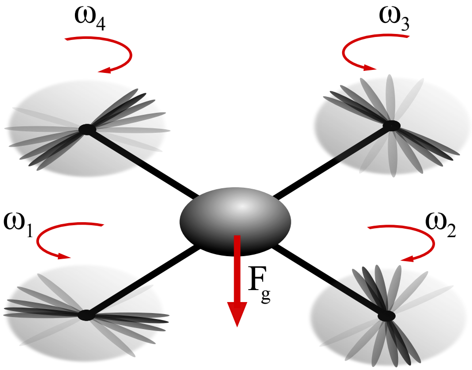
Model



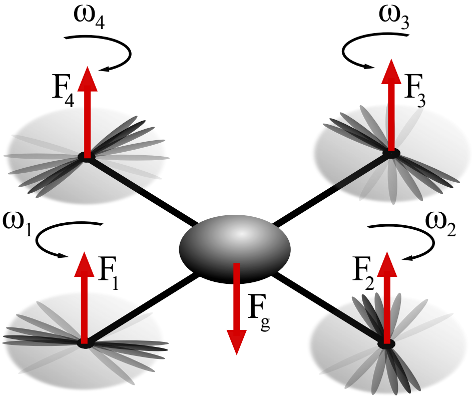
Model



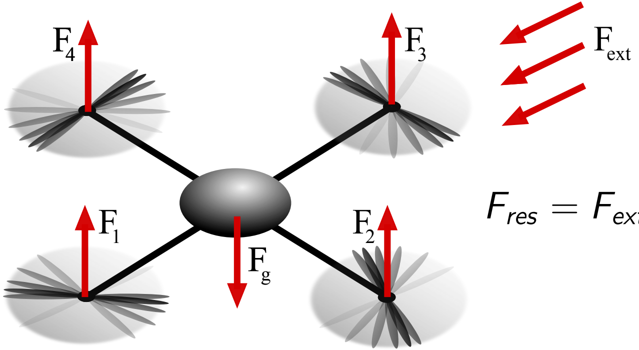
Model



Model

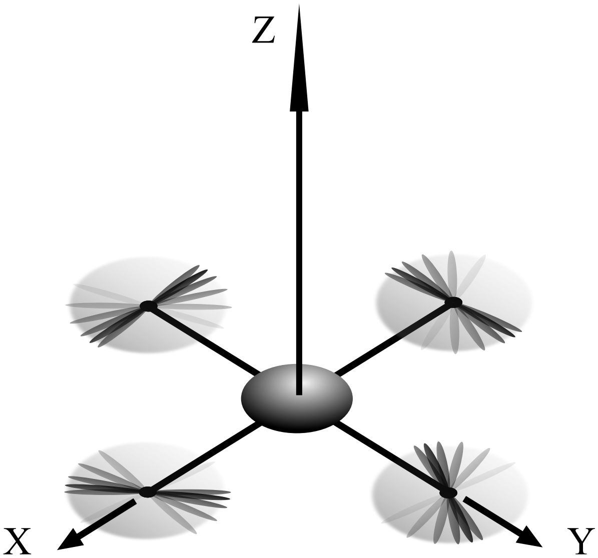


Forces

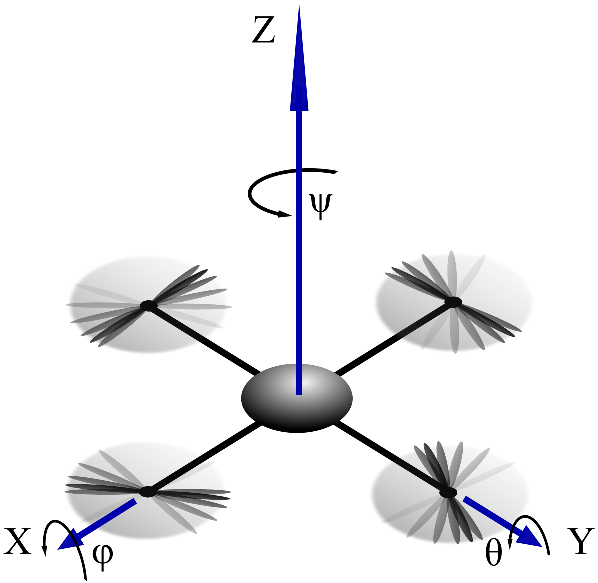


$$F_{res} = F_{ext} + F_g + \sum_{i=1}^{4} F_i$$

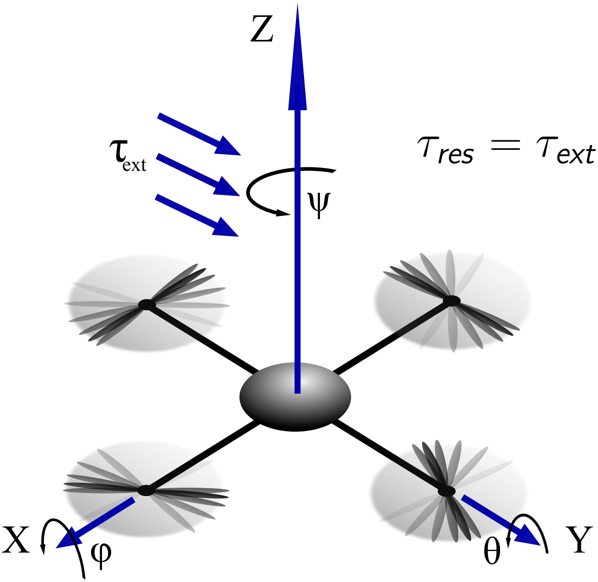
Torques



Torques



Torques



$$\tau_{res} = \tau_{ext} + \tau_{\psi} + \tau_{\varphi} + \tau_{\theta}$$

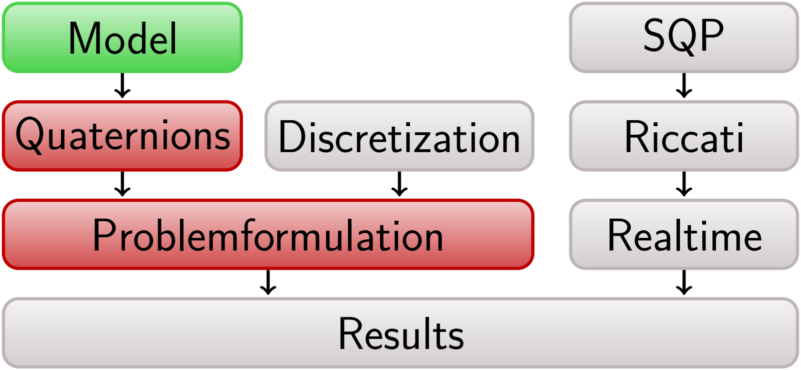
Obtain ODE

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

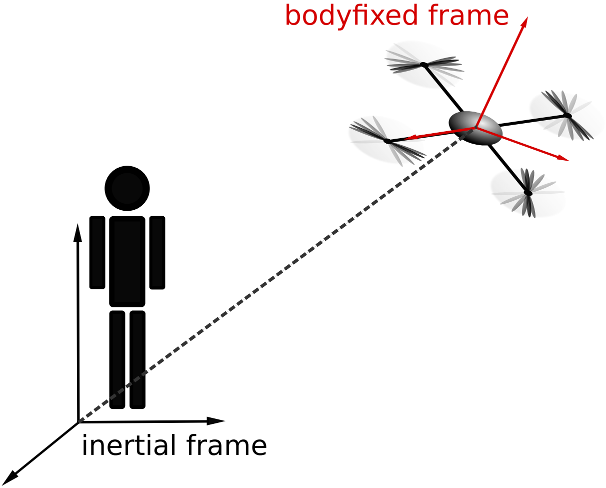
Obtain ODE

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

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


Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results

Copter Rotation



Quaternions

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

$$\Leftrightarrow$$

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

Quaternions

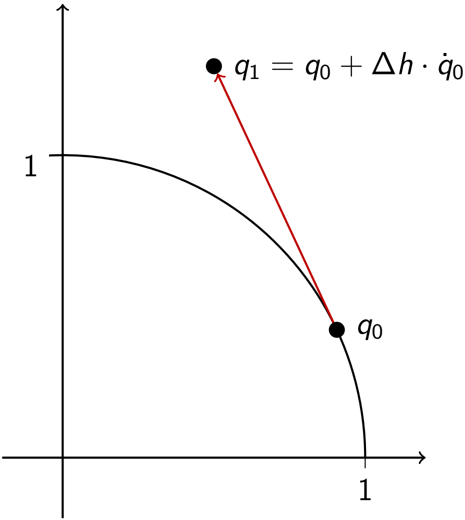
$$q = a + \mathrm{i}b + \mathrm{j}c + \mathrm{k}d \qquad a, b, c, d \in \mathbb{R}$$

$$\Leftrightarrow$$

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

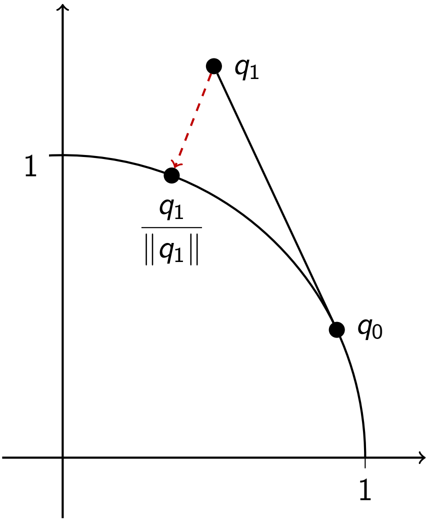
represent rotation $\quad \Leftrightarrow \quad \|q\| = 1 \quad \Leftrightarrow \quad q \in \mathcal{S}^3$

Drift Correction

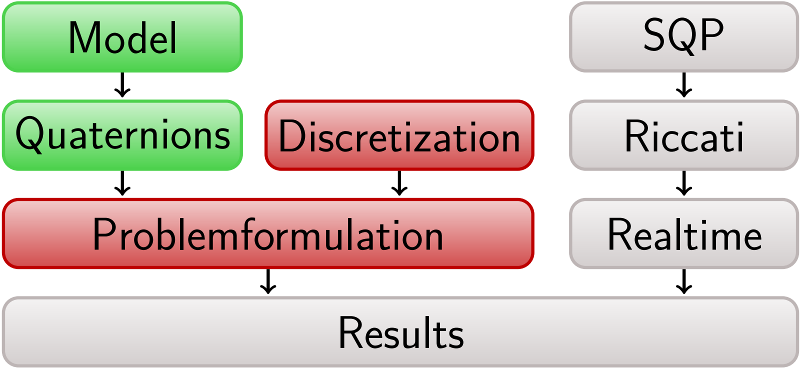


$$\dot{q}(t) = \tilde{f}(q(t))$$

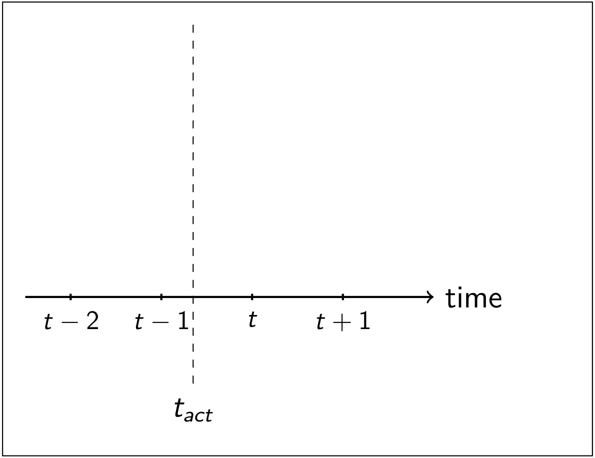
Drift Correction



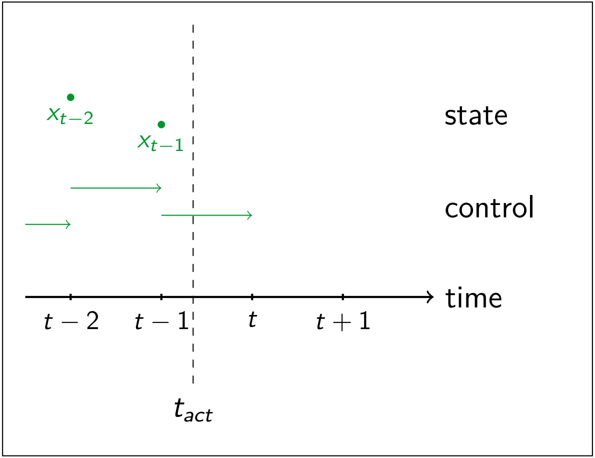
$$\dot{q}(t) = \tilde{f}(q(t)) - \lambda(q(t))$$


Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results

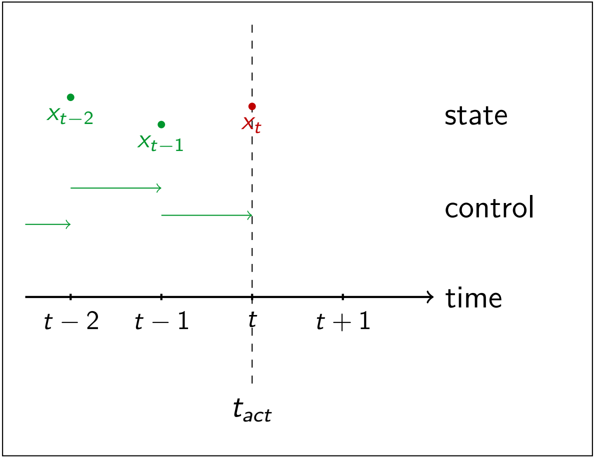
Setting



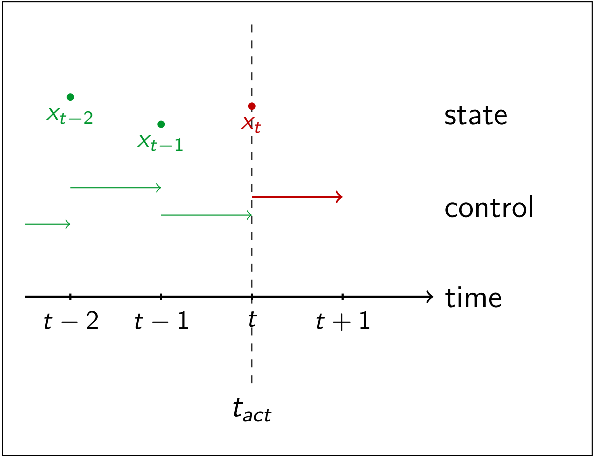
Setting



Setting



Setting

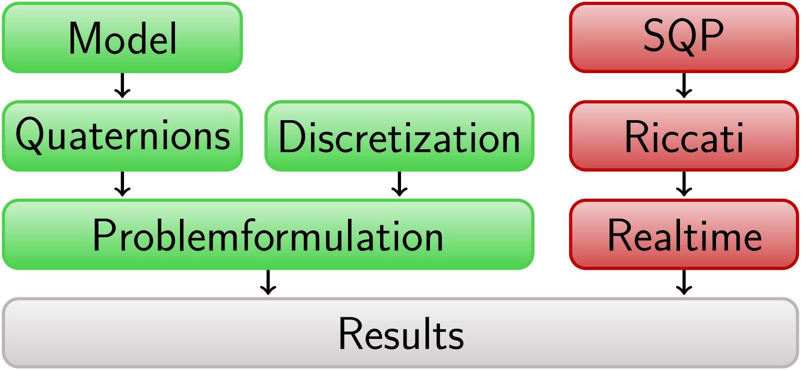


Discrete Problem

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

$J_i(x_i, u_i)$ discretized goal function

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


Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results

The Lagrangian

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

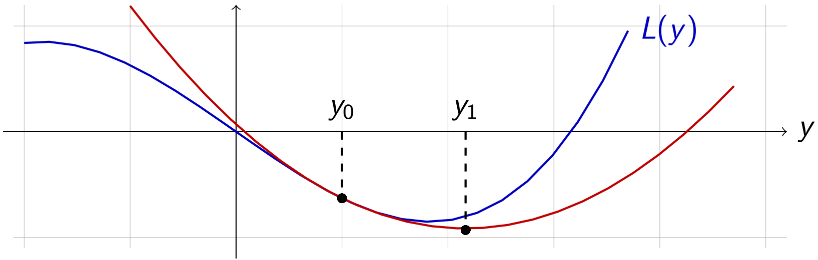
$$y := (\lambda, x, u)$$
$$y^* \text{ optimal} \quad \Leftrightarrow \quad \nabla_y L(y^*) = 0$$

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

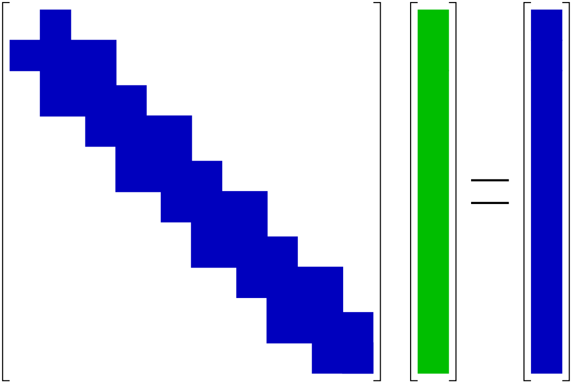
Find s with:

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

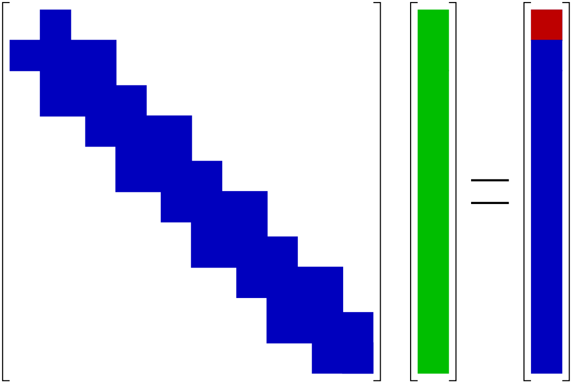
Approximate $\nabla^2 L(y_0)$ and solve:

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

Riccati Recursion

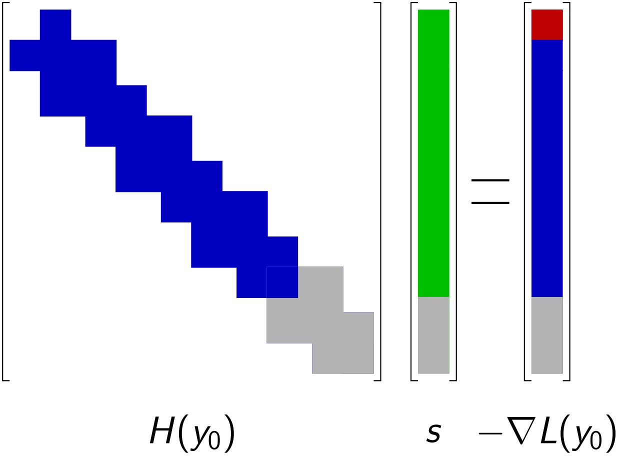


Riccati Recursion

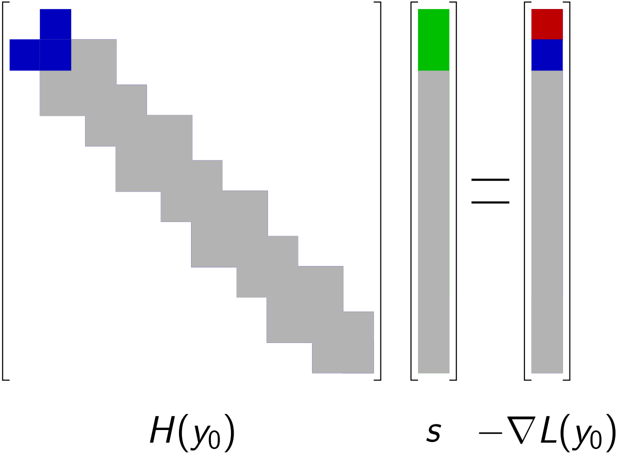


$H(y_0) \quad s \quad -\nabla L(y_0)$

Riccati Recursion

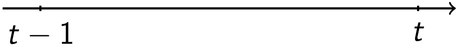


Riccati Recursion



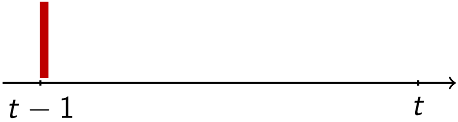
Summary

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



Summary

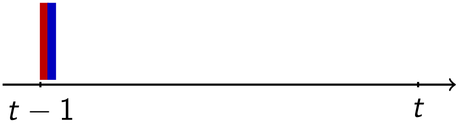
What happens in interval $[t-1, t]$?



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)

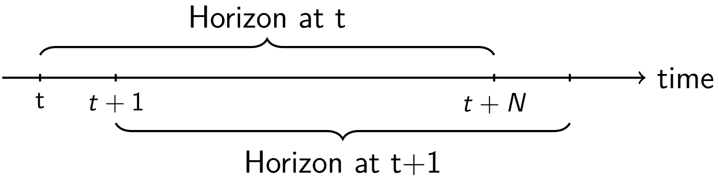
Summary

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)

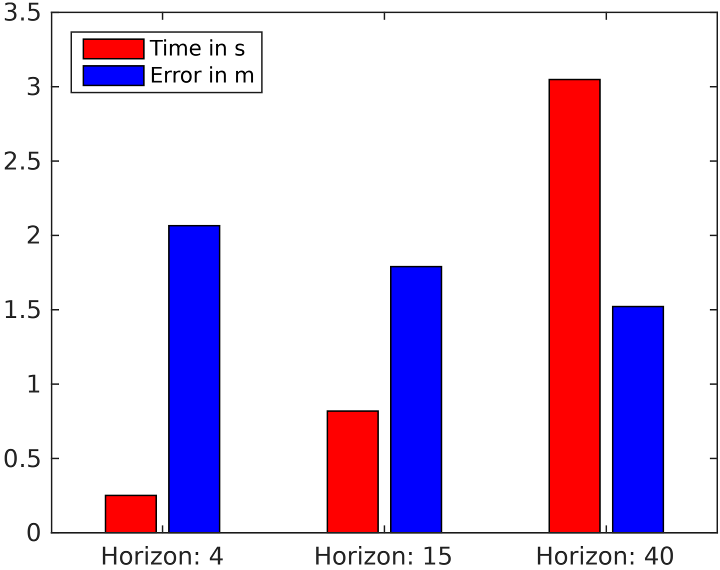
$\Rightarrow$ with horizon 18 this is 28% faster than backslash-operator

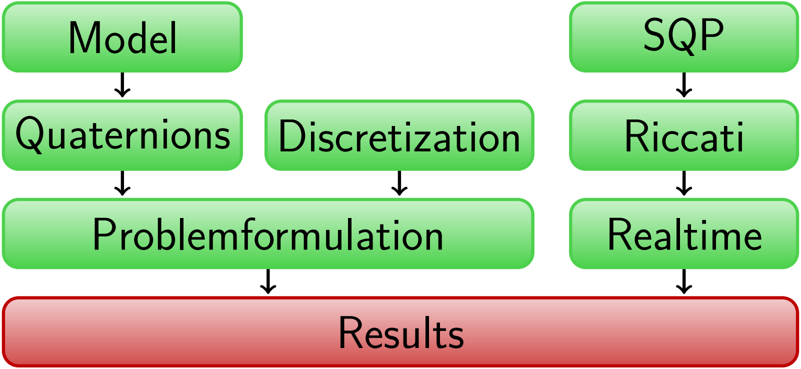
Finite Horizon



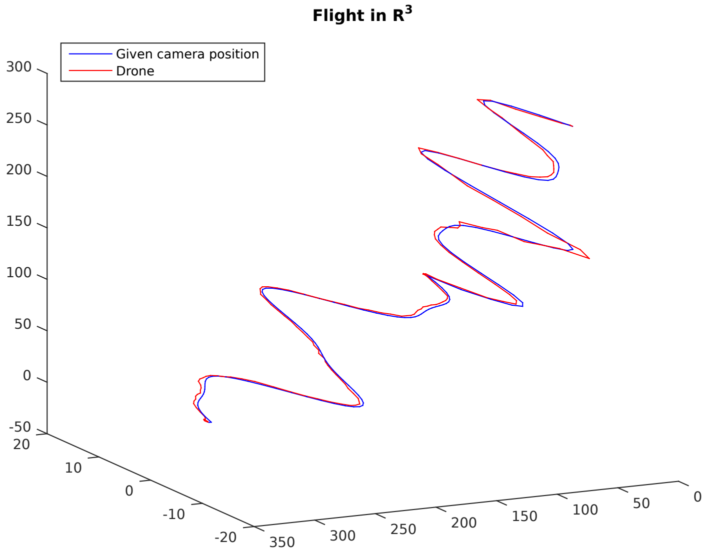
$\Rightarrow$ compromise between accuracy and speed

Finite Horizon

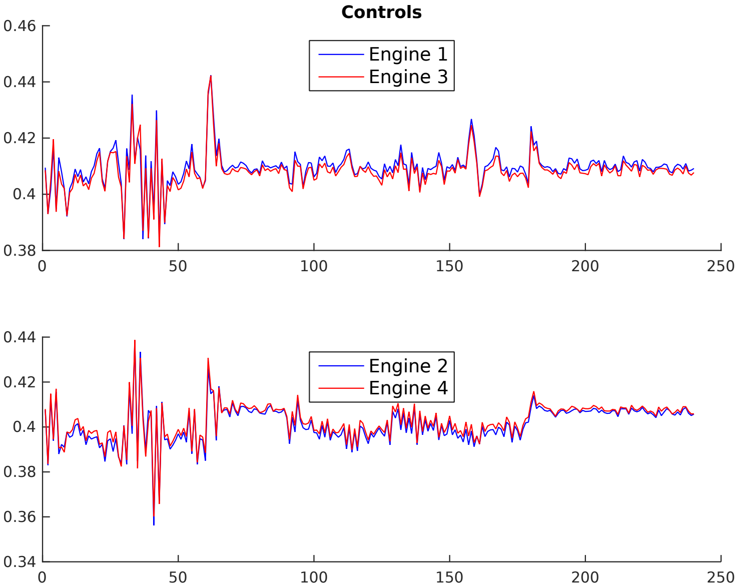



Model
SQP
Quaternions
Discretization
Riccati
Problemformulation
Realtime
Results

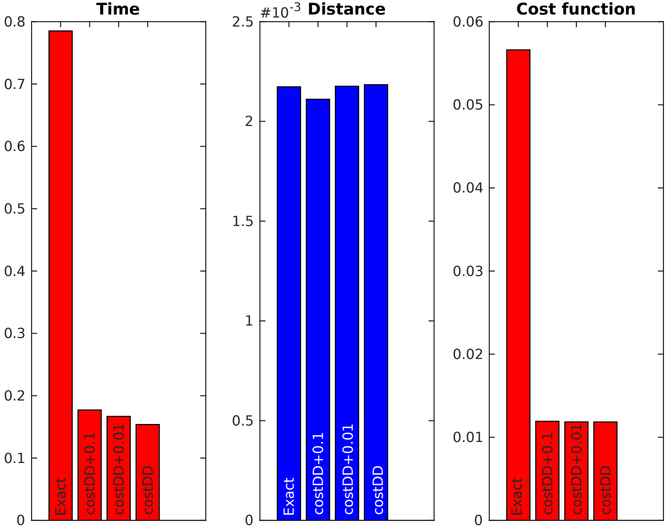
Following a Skier



Following a Skier



Approximations



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.
Real-Time Optimization for Large Scale Nonlinear Processes.
PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2001.

References II

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.

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.

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.
Geometriekalküle.
Springer: Berlin, Heidelberg, 2009.

What we learned from the Project:

- you have to know your plan to ignore it
- MATLAB® is special
- tests are helpful - or drive you crazy
- loopings can be cheap, too
- keep your colorscheme
- keep calm and do case studies

Any Questions?