10.3.2015

Optimal and Learning Control
for
Autonomous Robots
Lecture 4

Jonas Buchli
Agile & Dexterous Robotics Lab

Class logistics

Exercise 1 hand-out next week!

Script: LEE H 203 - Cornelia Della Casa (mornings)

Office hours: Thu, 17:30-18:30 Room: ML J37.1

First office hour March 5

Exercise groups

Sign up for the exercises **in groups of 2**:

https://ethz.doodle.com/c27fqrggtqth2x57

Please avoid single-member groups!

Erratum Script

p14

$$\frac{dV^*}{dt} = V_t^* + V_{\mathbf{x}}^{*T}\mathbf{f} + \frac{1}{2}\mathrm{Tr}\left[V_{\mathbf{xx}}^* E[(\mathbf{f}+\mathbf{Bw})(\mathbf{f}+\mathbf{Bw})^T]\Delta t\right]. \tag{1.55}$$

Lecture 4 Goals

- ★Sequential programming to solve nonlinear programming & optimal control problems
- ★Linear-quadratic assumption
- ★Derivation of ILQC (Part 1)

L3 Recap

Continuous time optimal control problem

Find control $u^*(t) = \mu^*(t, x(t))$ minimizing

control (input) policy

$$J = e^{-\beta(t_f - t_0)}\Phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} e^{-\beta(t-t_0)} L(\mathbf{x}(t), \mathbf{u}(t))dt$$

Given constraints

$$\dot{\mathbf{x}}(t) = \mathbf{f}_t(\mathbf{x}(t), \mathbf{u}(t))$$

Goal: Optimal policy

$$\mu^* = \arg\min_u J$$

Hamilton Jacobi Bellman Equation

Carl Gustav Jacob Jacobi
(1804-1851)

William Rowan Hamilton
(1805-1865)

Richard Bellman
1920-84

$$\boxed{\beta V^* - \frac{\partial V^*}{\partial t} = \min_{\mathbf{u} \in \mathbf{U}} \left\{ L(\mathbf{x}, \mathbf{u}) + \left(\frac{\partial V^*}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \right\}}$$

In general: Nonlinear, Partial Differential Equation
Has no analytical solution... : (

Backwards in time! $V^*(t_f, \mathbf{x}) = \Phi(\mathbf{x})$

Infinite time

$$J = \int_{t_0}^{\infty} e^{-\beta(t-t_0)} L(\mathbf{x}(t), \mathbf{u}(t)) dt$$

Value function not function of time: $\frac{\partial V^*}{\partial t} = 0$

$$\beta V^* = \min_{\mathbf{u} \in \mathbf{U}} \{ L(\mathbf{x}, \mathbf{u}) + \left(\frac{\partial V^*}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \}$$

Stochastic system

$$\dot{\mathbf{x}}(t) = \mathbf{f}_t(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{B}(t)\mathbf{w}(t), \quad \mathbf{x}(0) = \mathbf{x}_0$$

Mean: $E[\mathbf{w}(t)] = \bar{\mathbf{w}} = 0$ mean-free

Co-variance: $E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = \mathbf{W}(t)\delta(t-\tau)$ uncorrelated over time

$$E[\mathbf{w}(t)\mathbf{w}(\tau)^T] = 0$$
$$t \neq \tau$$

Expected cost:

$$J = E\left\{ e^{-\beta(t_f - t_0)}\Phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} e^{-\beta(t'-t_0)} L(\mathbf{x}(t'), \mathbf{u}(t'))dt' \right\}$$

'Stochastic' Hamilton Jacobi Bellman Equation

$$\beta V^*(t,\mathbf{x}) - V_t^*(t,\mathbf{x}) = \min_{\mathbf{u}(t)} \left\{ L(\mathbf{x},\mathbf{u}(t)) + V_{\mathbf{x}}^{*T} \mathbf{f}_t(\mathbf{x},\mathbf{u}(t)) + \boxed{\frac{1}{2}\mathbf{Tr}[V_{\mathbf{xx}}^* \mathbf{B}(t)\mathbf{W}(t)\mathbf{B}^T(t)]} \right\}$$

add'l cost

compare to deterministic HJB: $\beta V^* - \frac{\partial V^*}{\partial t} = \min_{\mathbf{u}\in\mathbf{U}} \left\{ L(\mathbf{x},\mathbf{u}) + \left(\frac{\partial V^*}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x},\mathbf{u}) \right\}$

Infinite time stochastic HJB

$$J = E\left\{\int_{t_0}^{\infty} e^{-\beta(t-t_0)} L(\mathbf{x}(t), \mathbf{u}(t))dt\right\}$$

$$\beta V^*(\mathbf{x}) = \min_{\mathbf{u}(t)}\left\{L(\mathbf{x}, \mathbf{u}(t)) + V^*_{\mathbf{x}}(x)\mathbf{f}_t(\mathbf{x}, \mathbf{u}(t)) + \frac{1}{2}\mathrm{Tr}[V^*_{\mathbf{xx}}\mathbf{B}(t)\mathbf{W}(t)\mathbf{B}^T(t)]\right\}$$

A D R L

ETH Zürich

EOF Recap

L4

today: iterative algorithm for discrete time problem

states \ time	discrete	continuous
discrete	gridworlds Markov chains	Petrinets Discrete Event Systems
continuous	$x_{n+1} = f(x_n, n)$ Map Difference equations $x_{n+1} = x_n + f_{\Delta}(x_n, n)$	Differential equations

‘Robots’
Newtonian Physics

Motivation: Example

[Mordatch et al, SIGGRAPH 12]

A D R L

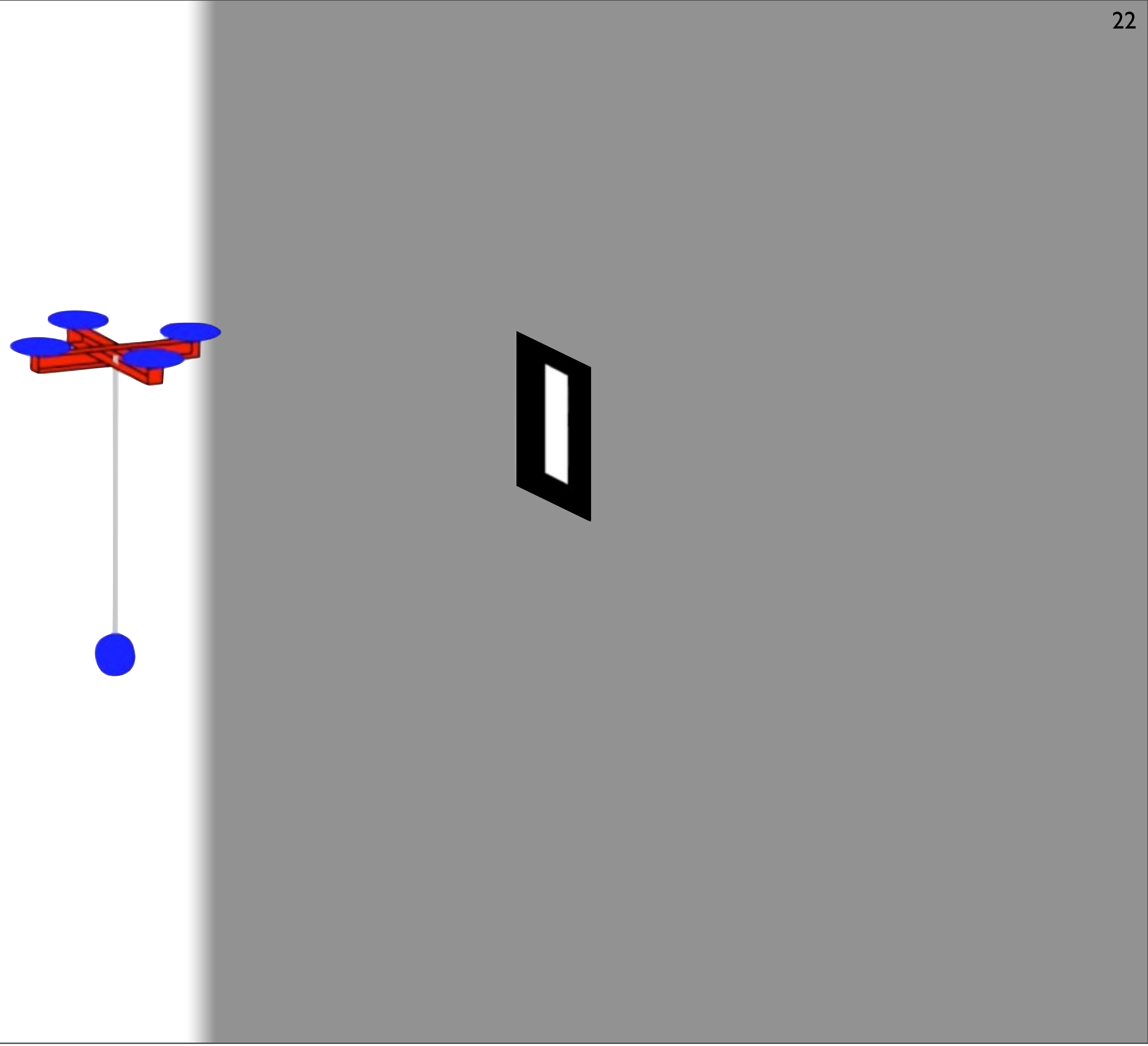
ETH Zürich

Rezero

http://rezero.ethz.ch

https://www.youtube.com/user/ADRLabETH

https://www.youtube.com/user/ADRLabETH



A D R L

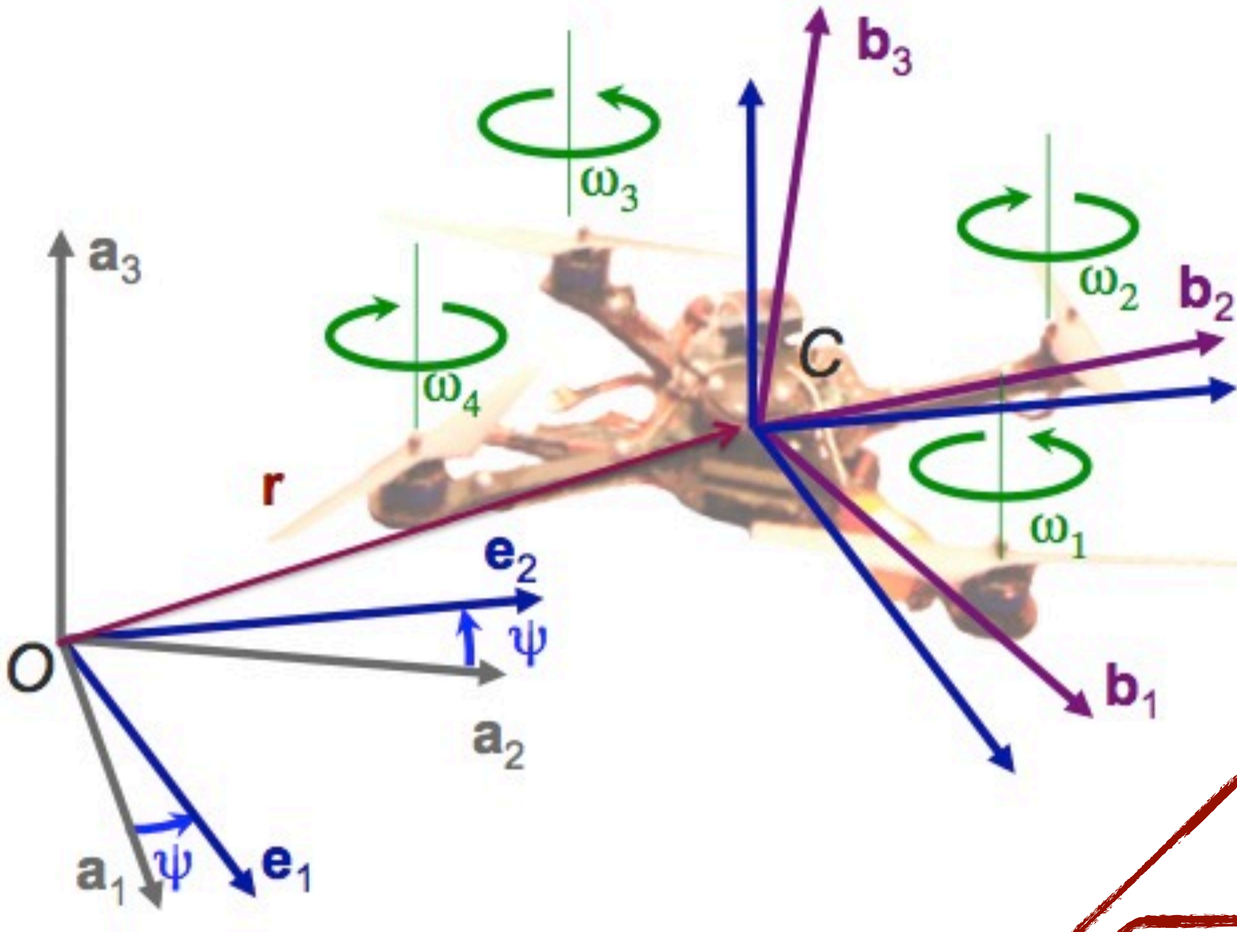
[de Crousaz, Farshidian, Buchli, ICRA 2015]

https://www.youtube.com/user/ADRLabETH

Unified Motion Control for Dynamic Quadrotor Maneuvers Demonstrated on Slung Load and Rotor Failure Tasks

C. de Crousaz, F. Farshidian, M. Neunert, J. Buchli
ICRA 2015

[de Crousaz, Farshidian, Buchli, ICRA 2015]



$$ {}^{\mathcal{A}}[R]_{\mathcal{B}} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} $$

$$ m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} $$

$$ I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} $$

★underactuated
★nonlinear

12 states

4 inputs

$$ \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} $$

Example from:
Kumar, Daniilidis, U Penn - MEAM 620: Robotics
https://alliance.seas.upenn.edu/~meam620

Iterative Optimal Control Algorithms

Solve optimal control problem

$$V^*(n,\mathbf{x}) = \min_{\mathbf{u}_n}\left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha V^*\left(n+1, \mathbf{f}_n\left(\mathbf{x},\mathbf{u}_n\right)\right)\right]$$

1. Principle of optimality: Bellman / HJB Equation
2. Solve for Value function
3. Calculate optimal value function
4. Compute optimal control from value function

... not tractable...
analytical solutions can't be found in general...!

Solve optimal control problem

$$V^*(n, \mathbf{x}) = \min_{\mathbf{u}_n} \left[L_n(\mathbf{x}, \mathbf{u}_n) + \alpha V^* \left(n+1, \mathbf{f}_n\left(\mathbf{x}, \mathbf{u}_n\right)\right)\right]$$

1. Principle of optimality: Bellman / HJB Equation
2. Make some assumptions
3. Minimize RHS of Equation
4. ... yields conditions for optimal control
5. substitute back to solve for remaining quantities

(... step back to function optimization ...

Nonlinear program

$$\min_x f(x) \qquad x \in \mathbb{R}^n$$

$$\begin{aligned} s.t. \qquad & f_j(x) \leq 0, \qquad & j = 1, \ldots, N \\ & h_j(x) = 0, \qquad & j = 1, \ldots, N \end{aligned}$$

f, h nonlinear
No analytical solution in general

NLP - Lagrangian

assuming only equality constraints

$$\min_x f(x) \qquad x \in \mathbb{R}^n$$

$$h_j(x) = 0,$$

$$L = f(x) + \sum \lambda_i h_i(x)$$

$$\nabla_x L = 0 \qquad \frac{\partial L}{\partial x_i} = 0$$

$$\nabla_\lambda L = 0 \qquad \frac{\partial L}{\partial \lambda_i} = 0$$

No analytical solution for general nonlinear f or h

Linear / Quadratic program

$$f(x) = a_2 x^2 + a_1 x + a_0$$

$$h(x) = b_1 x + b_0$$

$$\frac{\partial L}{\partial x} = 2a_2 x + a_1 + \lambda b_1$$

if f quadratic and h linear
optimization problem always has a solution

Sequential Quadratic Programming (SQP)

Idea: Approximate nonlinear program by a QP, solve iteratively

- Initial guess $\tilde{x}_0$
- Approximate $f(x)$ at $\tilde{x}_0$ by 2nd order Taylor series expansion

$$f(x) \approx f(\tilde{x}_0) + (x - \tilde{x}_0)^T \nabla f(\tilde{x}_0) + \frac{1}{2}(x - \tilde{x}_0)^T \nabla^2 f(\tilde{x}_0)(x - \tilde{x}_0)$$

square in x

$$f_j(x) \approx f_j(\tilde{x}_0) + (x - \tilde{x}_0)^T \nabla f_j(\tilde{x}_0)$$

$$h_j(x) \approx h_j(\tilde{x}_0) + (x - \tilde{x}_0)^T \nabla h_j(\tilde{x}_0)$$

constraints: first order

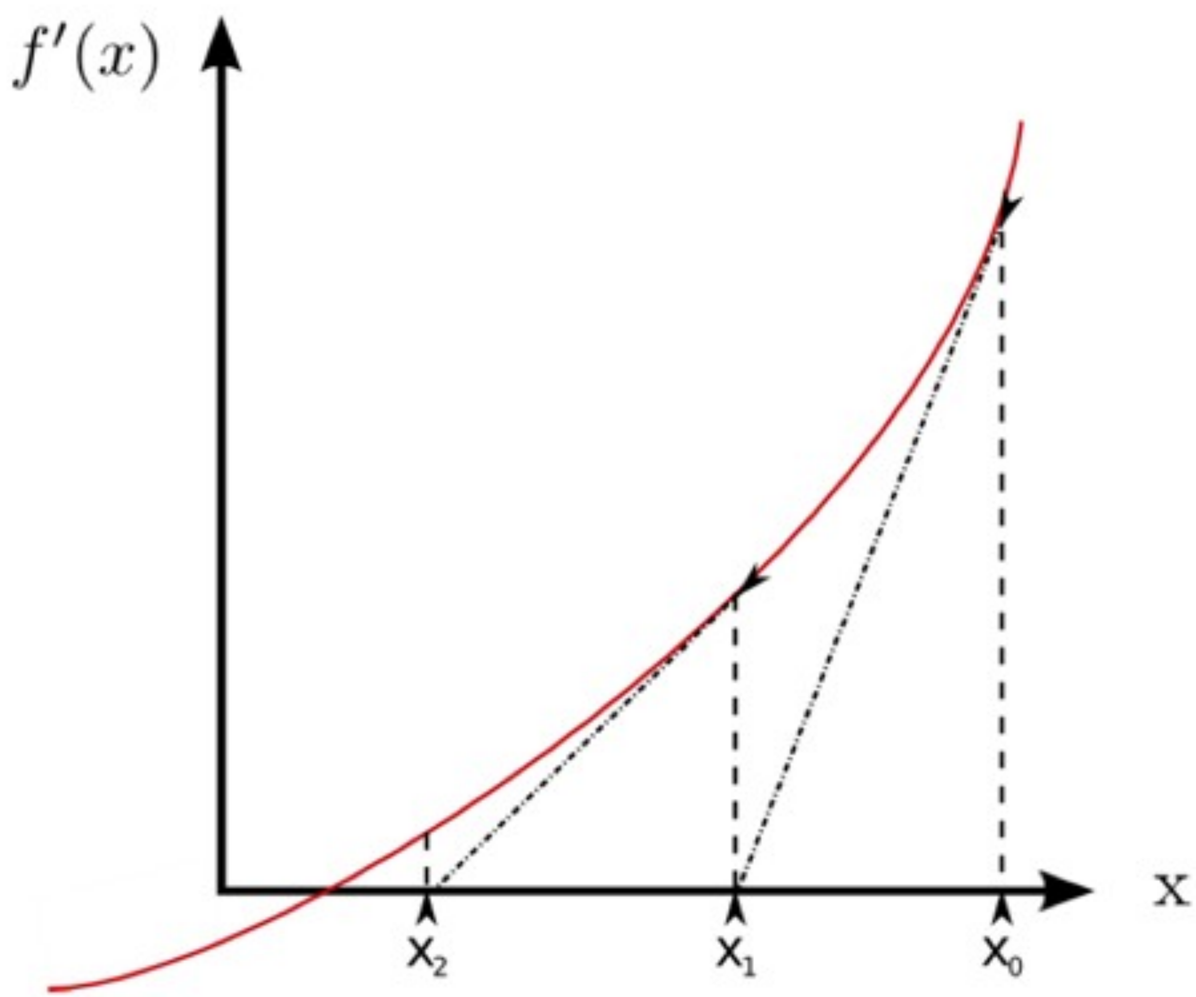
- yields new approximative solution $\tilde{x}_1$
- repeat

$$\lim_{i \to \infty} \tilde{x}_i = x^*$$

if problem convex

Newton Raphson

$$\tilde{x}_1 = \tilde{x}_0 - \frac{f'(\tilde{x}_0)}{f''(\tilde{x}_0)}$$



$$f(x) = 0.5ax^2 + bx + c \text{ ?}$$

f' is linear

$$x^* = \tilde{x}_0 - \frac{f'(\tilde{x}_0)}{f''(\tilde{x}_0)} = \tilde{x}_0 - \frac{a\tilde{x}_0 + b}{a} = -\frac{b}{a}$$

A D R L

ETH Zürich

... back to optimal control

Sequential Linear Quadratic Control - SLQ

$$\min_{\mu} \left[\Phi(\mathbf{x}(N)) + \sum_{n=0}^{N-1} L_n(\mathbf{x}(n), \mathbf{u}(n)) \right]$$

$$s.t. \qquad \mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n)) \qquad\qquad \mathbf{x}(0) = \mathbf{x}_0$$

$$\mathbf{u}(n, \mathbf{x}) = \mu(n, \mathbf{x})$$

Idea: Fit simplified subproblem to original problem, solve iteratively

Class of algorithms

SQP vs SLQ

SQP

1. Initial guess for parameter
2. Solve sub problem: Approximate original problem with a linear-quadratic problem
3. yields new approximative solution
4. repeat

SLQ

1. Initial guess for policy
2. Solve sub problem: Approximate value function with a linear-quadratic
3. yields new approximative policy
4. repeat

SLQ subproblem in a nutshell

2.1 Forward pass:

integrate to get a state (and controls) trajectory

2.2 Backward pass

Solve simplified optimal control problem around state and control trajectory

3. Adjust guess for optimal control

- choice of: approximation, solver ⇒different SLQ algorithms

(examples: DDP, iLQG, ILQC)

1. Guess an initial (stabilizing) control control policy $\mu^0(n, x)$
2. "Roll out": Apply the control policy to the non-linear system (1.68) (forward integration), which yields the state trajectory $\mathbf{X}_k = \{\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(N)\}$ and input trajectory $\mathbf{U}_k = \{\mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(N-1)\}$.
3. Starting with $n = N - 1$, approximate the value function as a quadratic function around the pair $(\mathbf{x}(N-1), \mathbf{u}(N-1))$.
4. Having a quadratic value function, the Bellman equation can be solved efficiently. The output of this step is a control policy at time $N - 1$ which minimizes the quadratic value function.
5. "Backward pass": Repeat steps 3 and 4 for every state-input pair along the trajectory yielding $\delta\mu^k = \{\bar{\mathbf{u}}(0, \mathbf{x}) \ldots, \bar{\mathbf{u}}(N-1, \mathbf{x})\}$. The updated optimized control inputs are then calculated with an an appropriate step-size α_k from

$$\mu^{k+1} = \mu^k + \alpha_k \cdot \delta\mu^k \tag{1.69}$$

6. Iterate through steps 2 → 5 using the updated control policy μ^{k+1} until a termination condition is satisfied, e.g. no more cost improvement or no more control vector changes.

Notice that if our system dynamics is already linear, and the cost function quadratic, then only one iteration step is necessary to find the globally optimal solution, similar to in (1.66). In this case the SLQ controller reduces to a LQR controller.

ILQC
Overview of derivation

Linearize system dynamics
Quadratize cost

Compute value function
Compute optimal control
Solve for Riccati like equation
Solve Riccati like equation

ILQC

Problem statement

$$\mathbf{x}_{n+1} = \mathbf{f}_n(\mathbf{x}_n, \mathbf{u}_n), \qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad n \in \{0, 1, ..., N-1\}$$

state-vector $\mathbf{x}_n$

control input vector $\mathbf{u}_n$

$$\alpha = 1$$

$$J = \Phi(\mathbf{x}_N) + \sum_{n=0}^{N-1} L_n(\mathbf{x}_n, \mathbf{u}_n)$$

$$\boldsymbol{\mu} = \{\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{N-1}\}$$

Goal: Find locally optimal control law $\boldsymbol{\mu}^*$

A D R L

ETH Zürich

ILQC

Local linear-quadratic approximation

stable control policy $\boldsymbol{\mu}(n, \mathbf{x})$

initial condition $\bar{\mathbf{x}}(0) = \mathbf{x}_0$

Forward integrate $\mathbf{x}_{n+1} = \mathbf{f}_n(\mathbf{x}_n, \mathbf{u}_n)$ using $\boldsymbol{\mu}$

yields state and control trajectory

$$\{\bar{\mathbf{x}}_n\} \qquad \{\bar{\mathbf{u}}_n\}$$

Linearize system dynamics, quadratize cost function around every pair: $(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)$

Small deviation from sampled x, u:

$$\delta \mathbf{x}_n \triangleq \mathbf{x}_n - \bar{\mathbf{x}}_n$$

$$\delta \mathbf{u}_n \triangleq \mathbf{u}_n - \bar{\mathbf{u}}_n$$

$$\delta \mathbf{x}(0) = 0$$

Linearization of system dynamics

$$\bar{\mathbf{x}}_{n+1} + \delta\mathbf{x}_{n+1} = \mathbf{f}_n(\bar{\mathbf{x}}_n + \delta\mathbf{x}_n, \bar{\mathbf{u}}_n + \delta\mathbf{u}_n)$$

$$\approx \boxed{\mathbf{f}_n(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)} + \boxed{\frac{\partial \mathbf{f}(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{x}}\delta\mathbf{x}_n + \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{u}}\delta\mathbf{u}_n}$$

$$\bar{\mathbf{x}}_{n+1} = \mathbf{f}_n(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)$$

$$\delta\mathbf{x}_{n+1} \approx \mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n$$

$$\mathbf{A}_n = \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{x}}$$ systems matrix

$$\mathbf{B}_n = \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{u}}$$ control gain matrix

$\mathbf{A}_n$ and $\mathbf{B}_n$ are independent of $\delta\mathbf{x}_n$ and $\delta\mathbf{u}_n$

$\mathbf{A}_n$ and $\mathbf{B}_n$ are time varying

nonlinear → linear, time variant

Quadratization of cost function

$$J = \Phi(\mathbf{x}_N) + \sum_{n=0}^{N-1} L_n(\mathbf{x}_n, \mathbf{u}_n)$$

$$J \approx q_N + \delta\mathbf{x}_N^T\mathbf{q}_N + \frac{1}{2}\delta\mathbf{x}_N^T\mathbf{Q}_N\delta\mathbf{x}_N$$
$$+ \sum_{n=0}^{N-1}\{q_n + \delta\mathbf{x}_n^T\mathbf{q}_n + \delta\mathbf{u}_n^T\mathbf{r}_n + \frac{1}{2}\delta\mathbf{x}_n^T\mathbf{Q}_n\delta\mathbf{x}_n + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{R}_n\delta\mathbf{u}_n + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n\}$$

Control costs

State costs

'Mixing terms'

$\forall n \in \{0, \cdots, N-1\}$:

$$q_n = L_n(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n), \qquad \mathbf{q}_n = \frac{\partial L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{x}}, \qquad \mathbf{Q}_n = \frac{\partial^2 L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{x}^2}$$

$$\mathbf{P}_n = \frac{\partial^2 L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{u}\partial \mathbf{x}}, \qquad \mathbf{r}_n = \frac{\partial L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{u}}, \qquad \mathbf{R}_n = \frac{\partial^2 L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial \mathbf{u}^2}$$

$n = N$:

$$q_N = \Phi(\bar{\mathbf{x}}_n), \qquad \mathbf{q}_N = \frac{\partial \Phi(\bar{\mathbf{x}}_n)}{\partial \mathbf{x}}, \qquad \mathbf{Q}_N = \frac{\partial^2 \Phi(\bar{\mathbf{x}}_n)}{\partial \mathbf{x}^2}$$

Note that all derivatives w.r.t. $\mathbf{u}$ are zero for the terminal time-step N

Q, R, P are given through definition of cost!

using the LQ assumptions we can now derive an approximately-optimal control law
for this need to compute value function... plug in...

Compute Value function

(1) Bellman Equation w. quadratic cost

Bellman Equation:

$$V^*(n, \delta\mathbf{x}_n) = \min_{\mathbf{u}_n} \left[L_n(\mathbf{x}_n, \mathbf{u}_n) + V^*(n+1, \delta\mathbf{x}_{n+1})\right]$$

Plug in quadratic cost (Slide 45):

$$V^*(n, \delta\mathbf{x}_n) = \min_{\mathbf{u}_n} \Big[q_n + \delta\mathbf{x}_n^T(\mathbf{q}_n + \frac{1}{2}\mathbf{Q}_n\delta\mathbf{x}_n) + \delta\mathbf{u}_n^T(\mathbf{r}_n + \frac{1}{2}\mathbf{R}_n\delta\mathbf{u}_n) + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n$$
$$+ V^*(n+1, \mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n) \Big]$$

A D R L

Compute Value function

(2) Quadratic Ansatz for Value function

Ansatz: Quadratic Value function

$$V^*(n+1, \delta\mathbf{x}_{n+1}) = s_{n+1} + \delta\mathbf{x}_{n+1}^T \mathbf{s}_{n+1} + \frac{1}{2}\delta\mathbf{x}_{n+1}^T \mathbf{S}_{n+1} \delta\mathbf{x}_{n+1}$$

$\mathbf{S}_n, \mathbf{s}_n, s_n$ are unknown

and will have to be computed... later!

Compute Value function

(3) Combine Ansatz and quadratic cost

Value function with quadratic cost (1):

$$V^*(n, \delta\mathbf{x}_n) = \min_{\mathbf{u}_n} \Big[q_n + \delta\mathbf{x}_n^T(\mathbf{q}_n + \frac{1}{2}\mathbf{Q}_n\delta\mathbf{x}_n) + \delta\mathbf{u}_n^T(\mathbf{r}_n + \frac{1}{2}\mathbf{R}_n\delta\mathbf{u}_n) + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n$$
$$+ V^*(n+1, \mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n) \Big]$$

Plug in Ansatz (2 - $V^*(n+1, \delta\mathbf{x}_{n+1}) = s_{n+1} + \delta\mathbf{x}_{n+1}^T\mathbf{s}_{n+1} + \frac{1}{2}\delta\mathbf{x}_{n+1}^T\mathbf{S}_{n+1}\delta\mathbf{x}_{n+1}$):

$$V(n+1, \mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n) =$$
$$s_{n+1} + (\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)\mathbf{s}_{n+1} + \frac{1}{2}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)^T\mathbf{S}_{n+1}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)$$

yields for RHS ($q_n + \delta\mathbf{x}_n^T(\mathbf{q}_n + \frac{1}{2}\mathbf{Q}_n\delta\mathbf{x}_n) + \delta\mathbf{u}_n^T(\mathbf{r}_n + \frac{1}{2}\mathbf{R}_n\delta\mathbf{u}_n) + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n + V^*(n+1, \mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)$)

$$\boxed{\begin{aligned} &q_n + \delta\mathbf{x}_n^T(\mathbf{q}_n + \frac{1}{2}\mathbf{Q}_n\delta\mathbf{x}_n) + \delta\mathbf{u}_n^T(\mathbf{r}_n + \frac{1}{2}\mathbf{R}_n\delta\mathbf{u}_n) + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n + \\ &s_{n+1} + (\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)\mathbf{s}_{n+1} + \frac{1}{2}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)^T\mathbf{S}_{n+1}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)\end{aligned}}$$

Compute Value function (4) rearrange

$$q_n + \delta\mathbf{x}_n^T(\mathbf{q}_n + \frac{1}{2}\mathbf{Q}_n\delta\mathbf{x}_n) + \delta\mathbf{u}_n^T(\mathbf{r}_n + \frac{1}{2}\mathbf{R}_n\delta\mathbf{u}_n) + \delta\mathbf{u}_n^T\mathbf{P}_n\delta\mathbf{x}_n +$$
$$s_{n+1} + (\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)\mathbf{s}_{n+1} + \frac{1}{2}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)^T\mathbf{S}_{n+1}(\mathbf{A}_n\delta\mathbf{x}_n + \mathbf{B}_n\delta\mathbf{u}_n)$$

'constants' $\quad q_n + s_{n+1} +$

'state dependent' $\quad \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1}) +$

'state-squared dependent' $\quad \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n +$

'control dependent' $\quad \delta\mathbf{u}_n^T(\mathbf{r}_n + \mathbf{B}_n^T\mathbf{s}_{n+1}) +$

'control-squared dependent' $\quad \frac{1}{2}\delta\mathbf{u}_n^T(\mathbf{R}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{B}_n)\delta\mathbf{u}_n +$

'mixed terms' $\quad \frac{1}{2}\delta\mathbf{u}_n^T(\mathbf{P}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n$

relabel terms depending on controls

$$\mathbf{g}_n \triangleq \mathbf{r}_n + \mathbf{B}_n^T\mathbf{s}_{n+1}$$
$$\mathbf{G}_n \triangleq \mathbf{P}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{A}_n$$
$$\mathbf{H}_n \triangleq \mathbf{R}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{B}_n$$

ETH zürich

relabel terms depending on controls

$$\mathbf{g}_n \triangleq \mathbf{r}_n + \mathbf{B}_n^T \mathbf{s}_{n+1}$$
$$\mathbf{G}_n \triangleq \mathbf{P}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{A}_n$$
$$\mathbf{H}_n \triangleq \mathbf{R}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{B}_n$$

$$V^*(n, \delta\mathbf{x}_n) = \min_{\mathbf{u}_n} \Big[q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T \mathbf{s}_{n+1}) + \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n + \delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n \Big] \qquad (1.80)$$

Find optimal control

$$V^*(n, \delta\mathbf{x}_n) = \min_{\mathbf{u}_n} \Big[q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1}) \tag{1.80}$$
$$+ \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n + \delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n \Big]$$

minimize RHS, set gradient in respect to controls = 0

$$\nabla_{\delta\mathbf{u}_n}\left[\delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n\right] = 0$$

$$\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n + \mathbf{H}_n\delta\mathbf{u}_n = 0$$

$$\boxed{\delta\mathbf{u}_n = -\mathbf{H}_n^{-1}\mathbf{g}_n - \mathbf{H}_n^{-1}\mathbf{G}_n\delta\mathbf{x}_n}$$

$$\mathbf{g}_n \triangleq \mathbf{r}_n + \mathbf{B}_n^T\mathbf{s}_{n+1}$$
$$\mathbf{G}_n \triangleq \mathbf{P}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{A}_n$$
$$\mathbf{H}_n \triangleq \mathbf{R}_n + \mathbf{B}_n^T\mathbf{S}_{n+1}\mathbf{B}_n$$

$$\mathbf{P}_n = \frac{\partial^2 L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial\mathbf{u}\partial\mathbf{x}}, \qquad \mathbf{r}_n = \frac{\partial L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial\mathbf{u}}, \qquad \mathbf{R}_n = \frac{\partial^2 L(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)}{\partial\mathbf{u}^2}$$

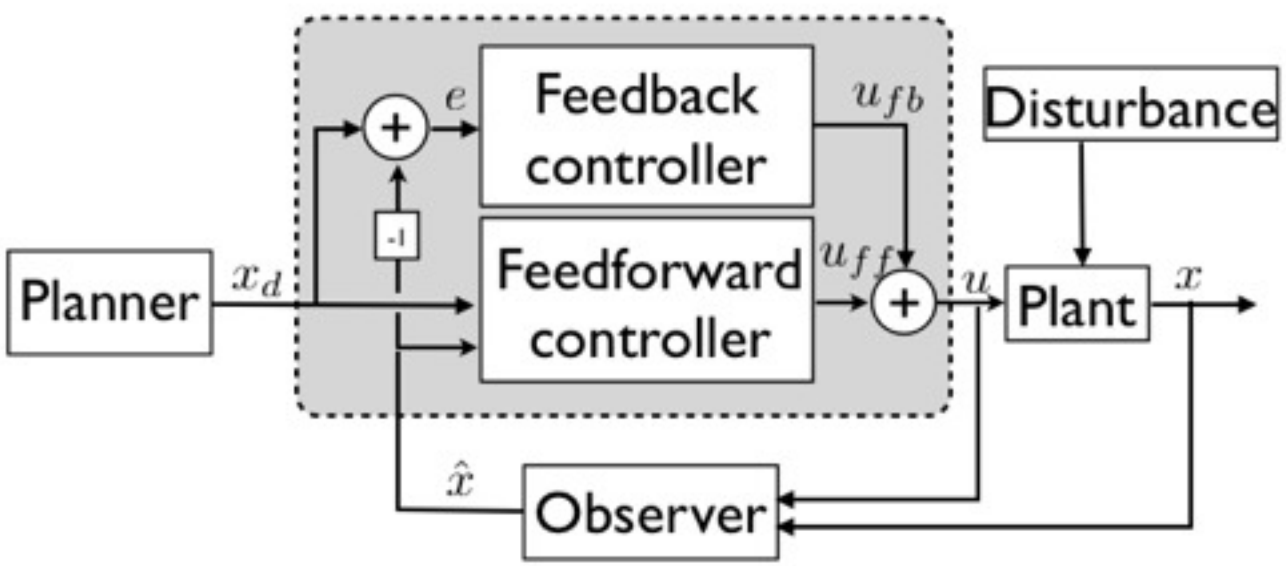
Optimal control: FF/FB

$$\delta \mathbf{u}_n = -\mathbf{H}_n^{-1}\mathbf{g}_n - \mathbf{H}_n^{-1}\mathbf{G}_n\delta\mathbf{x}_n$$

feed-forward term $\delta \mathbf{u}_n^{ff} = -\mathbf{H}_n^{-1}\mathbf{g}_n$

feedback term $\mathbf{K}_n\delta\mathbf{x}_n$ feedback gain matrix $\mathbf{K}_n := -\mathbf{H}_n^{-1}\mathbf{G}_n$

$$\delta \mathbf{u}_n = \delta \mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n$$



feed-forward term $\delta \mathbf{u}_n^{ff} = -\mathbf{H}_n^{-1}\mathbf{g}_n$

feedback gain matrix $\mathbf{K}_n := -\mathbf{H}_n^{-1}\mathbf{G}_n$

functions of unknown $\mathbf{S}_n, \mathbf{s}_n, s_n$

$$\boxed{\begin{aligned} \mathbf{g}_n &\triangleq \mathbf{r}_n + \mathbf{B}_n^T \mathbf{s}_{n+1} \\ \mathbf{G}_n &\triangleq \mathbf{P}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{A}_n \\ \mathbf{H}_n &\triangleq \mathbf{R}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{B}_n \end{aligned}}$$

Solving for $\mathbf{S}_n, \mathbf{s}_n, s_n$

L5 preview

Solving for $\mathbf{S}_n, \mathbf{s}_n, s_n$

$$\delta\mathbf{u}_n = -\mathbf{H}_n^{-1}\mathbf{g}_n - \mathbf{H}_n^{-1}\mathbf{G}_n\delta\mathbf{x}_n$$

feed-forward term $\delta\mathbf{u}_n^{ff} = -\mathbf{H}_n^{-1}\mathbf{g}_n$

feedback term $\mathbf{K}_n\delta\mathbf{x}_n$ feedback gain matrix $\mathbf{K}_n := -\mathbf{H}_n^{-1}\mathbf{G}_n$

replace $\delta\mathbf{u}_n = \delta\mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n$

plug into

$$V^*(n,\delta\mathbf{x}_n) = \min_{\mathbf{u}_n}\Big[q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1}) + \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n + \delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n\Big]$$

$$V^*(n,\delta\mathbf{x}_n) = \min_{\mathbf{u}_n}\Big[q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1})$$

$$+\frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n + \delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n\Big]$$

$$\boxed{\delta\mathbf{u}_n = \delta\mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n}$$

$$(\delta\mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n)^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n)$$
$$+\frac{1}{2}(\delta\mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n)^T\mathbf{H}_n(\delta\mathbf{u}_n^{ff} + \mathbf{K}_n\delta\mathbf{x}_n)$$

$$\delta\mathbf{u}_n^{ff^T}\mathbf{g}_n + \delta\mathbf{u}_n^{ff^T}\mathbf{G}_n\delta\mathbf{x}_n + \delta\mathbf{x}_n^T\mathbf{K}_n^T\mathbf{g}_n + \delta\mathbf{x}_n^T\mathbf{K}_n^T\mathbf{G}_n\delta\mathbf{x}_n$$
$$+\frac{1}{2}(\delta\mathbf{u}_n^{ff^T}\mathbf{H}_n\delta\mathbf{u}_n^{ff} + \delta\mathbf{u}_n^{ff^T}\mathbf{H}_n\mathbf{K}_n\delta\mathbf{x}_n + \delta\mathbf{x}_n^T\mathbf{K}_n^T\mathbf{H}_n\delta\mathbf{u}_n^{ff} + \delta\mathbf{x}_n^T\mathbf{K}_n^T\mathbf{H}_n\mathbf{K}_n\delta\mathbf{x}_n)$$

$$\delta\mathbf{u}^{ff^T}\mathbf{g} + \frac{1}{2}\delta\mathbf{u}^{ff^T}\mathbf{H}\delta\mathbf{u}^{ff} + \delta\mathbf{x}^T(\mathbf{G}^T\delta\mathbf{u}^{ff} + \mathbf{K}^T\mathbf{g} + \mathbf{K}^T\mathbf{H}\delta\mathbf{u}^{ff})$$
$$+\frac{1}{2}\delta\mathbf{x}^T(\mathbf{K}^T\mathbf{H}\mathbf{K} + \mathbf{K}^T\mathbf{G} + \mathbf{G}^T\mathbf{K})\delta\mathbf{x}$$

$$\boxed{V^*(n,\delta\mathbf{x}_n)} = \min_{\mathbf{u}_n}\Big[q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1})$$
$$+\frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n + \delta\mathbf{u}_n^T(\mathbf{g}_n + \mathbf{G}_n\delta\mathbf{x}_n) + \frac{1}{2}\delta\mathbf{u}_n^T\mathbf{H}_n\delta\mathbf{u}_n\Big]$$

$$\boxed{V^*(n,\delta\mathbf{x}_n) = s_n + \delta\mathbf{x}_n^T\mathbf{s}_n + \frac{1}{2}\delta\mathbf{x}_n^T\mathbf{S}_n\delta\mathbf{x}_n}$$

Quadratic Ansatz

Optimal control

$$\delta\mathbf{u}^{ff^T}\mathbf{g} + \frac{1}{2}\delta\mathbf{u}^{ff^T}\mathbf{H}\delta\mathbf{u}^{ff} + \delta\mathbf{x}^T(\mathbf{G}^T\delta\mathbf{u}^{ff} + \mathbf{K}^T\mathbf{g} + \mathbf{K}^T\mathbf{H}\delta\mathbf{u}^{ff})$$
$$+\frac{1}{2}\delta\mathbf{x}^T(\mathbf{K}^T\mathbf{H}\mathbf{K} + \mathbf{K}^T\mathbf{G} + \mathbf{G}^T\mathbf{K})\delta\mathbf{x}$$

$$s_n + \delta\mathbf{x}_n^T\mathbf{s}_n + \frac{1}{2}\delta\mathbf{x}_n^T\mathbf{S}_n\delta\mathbf{x}_n =$$
$$q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1}) + \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n +$$
$$\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}^T}\mathbf{g}_n + \frac{1}{2}\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}^T}\mathbf{H}_n\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}} + \delta\mathbf{x}_n^T(\mathbf{G}_n^T\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}} + \mathbf{K}_n^T\mathbf{g}_n + \mathbf{K}_n^T\mathbf{H}_n\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}})$$
$$+\frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{K}_n^T\mathbf{H}_n\mathbf{K}_n + \mathbf{K}_n^T\mathbf{G}_n + \mathbf{G}_n^T\mathbf{K}_n)\delta\mathbf{x}_n$$

sort into terms in $\delta\mathbf{x}^a \quad a \in [0,1,2]$

$1, \quad \delta\mathbf{x}, \quad \delta\mathbf{x}^T\delta\mathbf{x}$

sort into terms in $1,\ \delta\mathbf{x},\ \delta\mathbf{x}^T\delta\mathbf{x}^T$

$$s_n + \delta\mathbf{x}_n^T\mathbf{s}_n + \frac{1}{2}\delta\mathbf{x}_n^T\mathbf{S}_n\delta\mathbf{x}_n =$$

$$q_n + s_{n+1} + \delta\mathbf{x}_n^T(\mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1}) + \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n)\delta\mathbf{x}_n +$$

$$\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}\,T}\mathbf{g}_n + \frac{1}{2}\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}\,T}\mathbf{H}_n\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}} + \delta\mathbf{x}_n^T(\mathbf{G}_n^T\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}} + \mathbf{K}_n^T\mathbf{g}_n + \mathbf{K}_n^T\mathbf{H}_n\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}})$$

$$+ \frac{1}{2}\delta\mathbf{x}_n^T(\mathbf{K}_n^T\mathbf{H}_n\mathbf{K}_n + \mathbf{K}_n^T\mathbf{G}_n + \mathbf{G}_n^T\mathbf{K}_n)\delta\mathbf{x}_n$$

$$n \in \{0, \cdots, N-1\}$$

$$\mathbf{S}_n = \mathbf{Q}_n + \mathbf{A}_n^T\mathbf{S}_{n+1}\mathbf{A}_n + \mathbf{K}_n^T\mathbf{H}_n\mathbf{K}_n + \mathbf{K}_n^T\mathbf{G}_n + \mathbf{G}_n^T\mathbf{K}_n$$

$$\mathbf{s}_n = \mathbf{q}_n + \mathbf{A}_n^T\mathbf{s}_{n+1} + \mathbf{K}_n^T\mathbf{H}_n\delta\mathbf{u}_n^{ff} + \mathbf{K}_n^T\mathbf{g}_n + \mathbf{G}_n^T\delta\mathbf{u}_n^{ff}$$

$$s_n = q_n + s_{n+1} + \frac{1}{2}\delta\mathbf{u}_\mathbf{n}^{\mathbf{ff}\,T}\mathbf{H}_n\delta\mathbf{u}_n^{ff} + \delta\mathbf{u}_n^{ff\,T}\mathbf{g}_n$$

$$\mathbf{S}_N = \mathbf{Q}_N, \qquad \mathbf{s}_N = \mathbf{q}_N, \qquad s_N = q_N$$

A D R L

ETH Zürich

note symmetry of S (if Q symmetric)!
S positive definite

$$n \in \{0, \cdots, N-1\}$$

$$\mathbf{S}_n = \mathbf{Q}_n + \mathbf{A}_n^T \mathbf{S}_{n+1} \mathbf{A}_n + \mathbf{K}_n^T \mathbf{H}_n \mathbf{K}_n + \mathbf{K}_n^T \mathbf{G}_n + \mathbf{G}_n^T \mathbf{K}_n$$

$$\mathbf{s}_n = \mathbf{q}_n + \mathbf{A}_n^T \mathbf{s}_{n+1} + \mathbf{K}_n^T \mathbf{H}_n \delta \mathbf{u}_n^{ff} + \mathbf{K}_n^T \mathbf{g}_n + \mathbf{G}_n^T \delta \mathbf{u}_n^{ff}$$

$$s_n = q_n + s_{n+1} + \frac{1}{2} \delta \mathbf{u}_{\mathbf{n}}^{\mathbf{ff}^T} \mathbf{H}_n \delta \mathbf{u}_n^{ff} + \delta \mathbf{u}_n^{ff^T} \mathbf{g}_n$$

$$\mathbf{g}_n \triangleq \mathbf{r}_n + \mathbf{B}_n^T \mathbf{s}_{n+1}$$
$$\mathbf{G}_n \triangleq \mathbf{P}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{A}_n$$
$$\mathbf{H}_n \triangleq \mathbf{R}_n + \mathbf{B}_n^T \mathbf{S}_{n+1} \mathbf{B}_n$$

$$\mathbf{S}_N = \mathbf{Q}_N, \qquad \mathbf{s}_N = \mathbf{q}_N, \qquad s_N = q_N$$

- ★ S(n) are only a function of known quantities: system matrix, control gain matrix, cost terms
- ★ ...AND future S (backwards)
- ★ Principle of optimality: solve backwards in time

Optimal control

$$(\bar{\mathbf{x}}_n, \bar{\mathbf{u}}_n)$$

$$\delta \mathbf{x}_n \triangleq \mathbf{x}_n - \bar{\mathbf{x}}_n$$

$$\delta \mathbf{u}_n \triangleq \mathbf{u}_n - \bar{\mathbf{u}}_n$$

We have derived the ‘incremental’ policy, thus total control is

$$\mathbf{u}(n, x) = \bar{\mathbf{u}}_n + \delta \mathbf{u}_n^{ff} + \mathbf{K}_n(\mathbf{x}_n - \bar{\mathbf{x}}_n)$$

ILQGC main iteration

0. *Initialization:* we assume that an initial, feasible policy $\boldsymbol{\mu}$ and initial state $\mathbf{x}_0$ is given. Then, for every iteration (i):

1. *Roll-Out:* perform a forward-integration of the system dynamics (1.70) subject to initial condition $\mathbf{x}_0$ and the current policy $\boldsymbol{\mu}$. Thus, obtain the nominal state- and control input trajectories $\bar{\mathbf{u}}_n^{(i)}, \bar{\mathbf{x}}_n^{(i)}$ for $n = 0, 1, \ldots, N$.

2. *Linear-Quadratic Approximation:* build a local, linear-quadratic approximation around every state-input pair $(\bar{\mathbf{u}}_n^{(i)}, \bar{\mathbf{x}}_n^{(i)})$ as described in Equations (1.75) to (1.78).

3. *Compute the Control Law:* solve equations (1.84) to (1.86) backward in time and design the affine control policy through equation (1.88).

4. Go back to 1. and repeat until the sequences $\bar{\mathbf{u}}^{(i+1)}$ and $\bar{\mathbf{u}}^{(i)}$ are sufficiently close.

Credits

Quadcopter dynamics and illustrations:

Vijay Kumar, Kostas Daniilidis
U Penn - MEAM 620: Robotics
https://alliance.seas.upenn.edu/~meam620