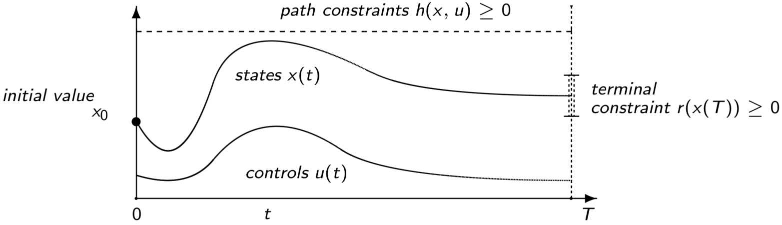
Optimal Control, MPC and MHE

Mario Zanon

Outline

Optimal Control Problem (OCP)



$$\underset{x(\cdot),u(\cdot),T}{\text{minimise}} \int_0^T \ell(x(\tau), u(\tau))\,\mathrm{d}\tau + V(x(T))$$

$$\begin{aligned} \text{subject to } \dot{x}(\tau) &= f(x(\tau), u(\tau)), & t \in [0, T], \quad & \text{(dynamic model)} \\ h(x(\tau), u(\tau)) &\geq 0, & t \in [0, T], \quad & \text{(path constraints)} \\ r(x(0), x(T)) &\geq 0. & & \text{(initial/terminal conditions)} \end{aligned}$$

OCP Solution Approacehs

Three families of approaches:

OCP Solution Approacehs

Three families of approaches:

- Hamilton-Jacobi-Bellman / Dynamic Programming

OCP Solution Approacehs

Three families of approaches:

- Hamilton-Jacobi-Bellman / Dynamic Programming
- Indirect Methods (first optimise then discretise)

OCP Solution Approacehs

Three families of approaches:

- Hamilton-Jacobi-Bellman / Dynamic Programming
- Indirect Methods (first optimise then discretise)
- Direct Methods (first discretise then optimise)

OCP Solution Approacehs

Three families of approaches:

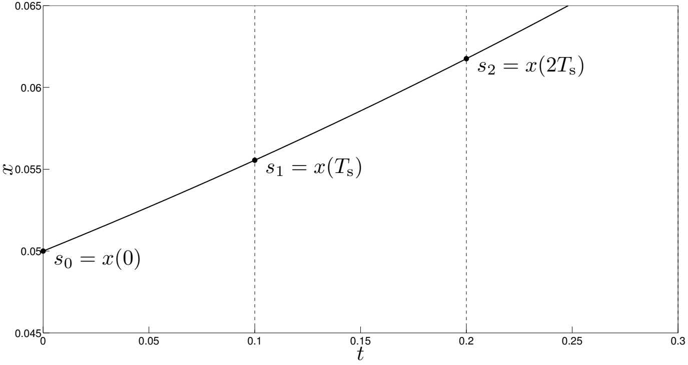
- Hamilton-Jacobi-Bellman / Dynamic Programming
- Indirect Methods (first optimise then discretise)
- **Direct Methods** (first discretise then optimise)
  - Direct single shooting
  - Direct multiple shooting
  - Direct collocation

Single vs. Multiple Shooting

Single vs. Multiple Shooting

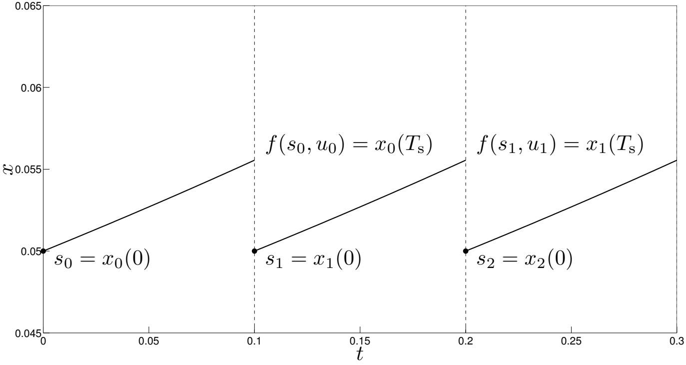
- Single Shooting:

From $x(t_0)$ integrate the system on **the whole horizon**
$\rightarrow$ **continuous trajectory**



Single vs. Multiple Shooting

- Single Shooting:
  From $x(t_0)$ integrate the system on **the whole horizon**
  $\rightarrow$ **continuous trajectory**
- Multiple Shooting:
  From $x(t_k)$ integrate the system on **each interval separately**
  $\rightarrow$ **discontinuous trajectory**



Single vs. Multiple Shooting

- Single Shooting:
  From $x(t_0)$ integrate the system on **the whole horizon**
  $\rightarrow$ **continuous trajectory**
- Multiple Shooting:
  From $x(t_k)$ integrate the system on **each interval separately**
  $\rightarrow$ **discontinuous trajectory**
- Colloctaion:
  For each point $x(t_k)$ satisfy the **collocation equations**
  $\rightarrow$ **discontinuous trajectory**

MS OCP Discretisation

The OCP becomes a nonlinear programming problem (NLP)

MS OCP Discretisation

The OCP becomes a nonlinear programming problem (NLP)

$$\underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}}{\text{minimise}} \quad \sum_{k=0}^{N} \ell(x_k, u_k) + V(x_N)$$

$$\begin{aligned}
\text{subject to} \quad & x_{k+1} = f(x_k, u_k), && k = 0, \dots, N-1, && \text{(dynamic model)} \\
& h(x_k, u_k) \geq 0, && k = 0, \dots, N-1, && \text{(path constraints)} \\
& r(x_0, x_N) \geq 0. && && \text{(initial/terminal conditions)}
\end{aligned}$$

MS OCP Discretisation

The OCP becomes a nonlinear programming problem (NLP)

$$\underset{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}}{\text{minimise}} \quad \sum_{k=0}^{N} \ell(x_k, u_k) + V(x_N)$$

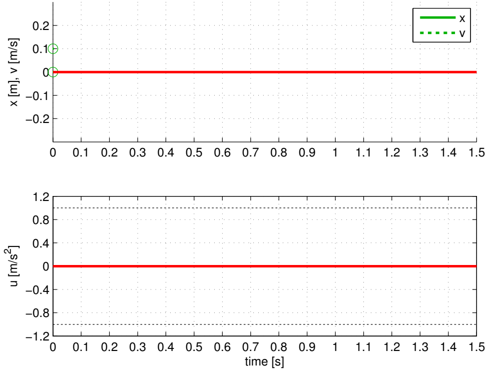
$$\begin{aligned} \text{subject to} \quad & x_{k+1} = f(x_k, u_k), && k = 0, \ldots, N-1, && \text{(dynamic model)} \\ & h(x_k, u_k) \geq 0, && k = 0, \ldots, N-1, && \text{(path constraints)} \\ & r(x_0, x_N) \geq 0. && && \text{(initial/terminal conditions)} \end{aligned}$$

... and it can be seen as an **OCP for discrete time systems**

Model Predictive Control

Looking ahead in the future...

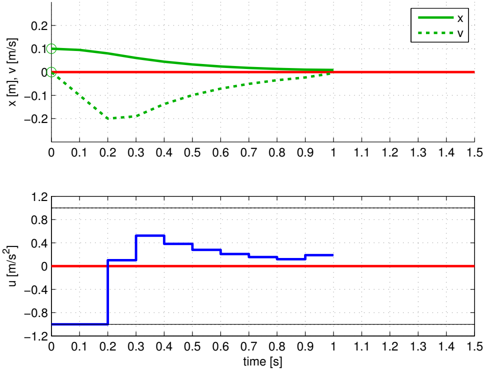
...design a control sequence minimizing the deviation from the reference



$$
\begin{aligned}
\min_{u,s} \quad & \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2 && \rightarrow \textbf{deviation} \text{ from the reference} \\
\text{s.t.} \quad & s_{k+1} = f\left(s_k, u_k\right), \quad k = 0, \dots, N-1, && \rightarrow \textbf{model} \text{ of the system evolution} \\
& h\left(s_k, u_k\right) \leq 0, \quad k = 0, \dots, N-1, && \rightarrow \textbf{path constraints} \\
& s_0 = \hat{x}_0. && \rightarrow \textbf{current state} \text{ of the system}
\end{aligned}
$$

Looking ahead in the future...

...design a control sequence minimizing the deviation from the reference



$$\min_{u,s} \quad \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2$$ → **deviation** from the reference

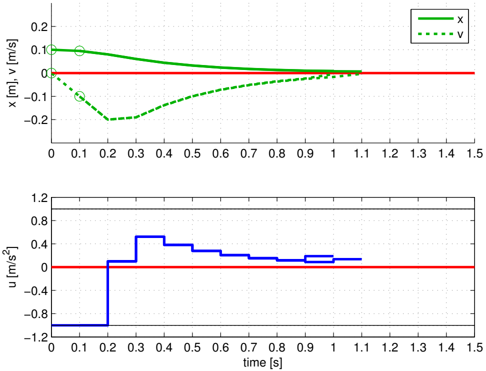
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k), \qquad k = 0, \dots, N-1,$$ → **model** of the system evolution

$$h(s_k, u_k) \leq 0, \qquad k = 0, \dots, N-1,$$ → **path constraints**

$$s_0 = \hat{x}_0.$$ → **current state** of the system

Looking ahead in the future...

...design a control sequence minimizing the deviation from the reference



$$\min_{u,s} \quad \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2 \quad \rightarrow \quad \textbf{deviation} \text{ from the reference}$$

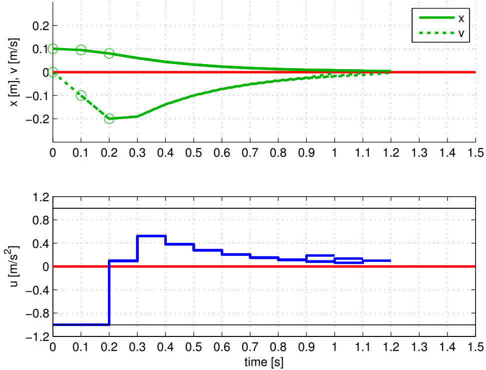
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k), \quad k = 0, \dots, N-1, \quad \rightarrow \quad \textbf{model} \text{ of the system evolution}$$

$$h(s_k, u_k) \le 0, \quad k = 0, \dots, N-1, \quad \rightarrow \quad \textbf{path constraints}$$

$$s_0 = \hat{x}_0. \quad \rightarrow \quad \textbf{current state} \text{ of the system}$$

Looking ahead in the future...

...design a control sequence minimizing the deviation from the reference



$$\min_{u,s} \quad \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2 \quad \rightarrow$$ **deviation** from the reference

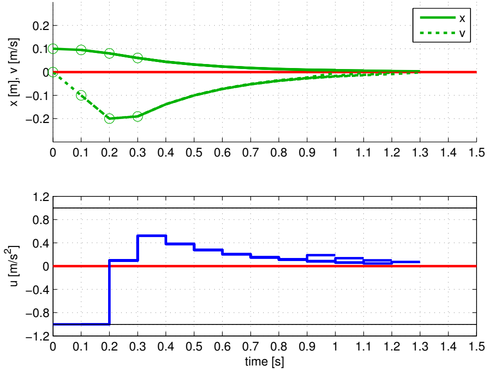
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k), \qquad k = 0, \ldots, N-1, \quad \rightarrow$$ **model** of the system evolution

$$h(s_k, u_k) \leq 0, \qquad k = 0, \ldots, N-1, \quad \rightarrow$$ **path constraints**

$$s_0 = \hat{x}_0. \quad \rightarrow$$ **current state** of the system

Looking ahead in the future...

...design a control sequence minimizing the deviation from the reference



$$\min_{u,s} \quad \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2$$ → **deviation** from the reference

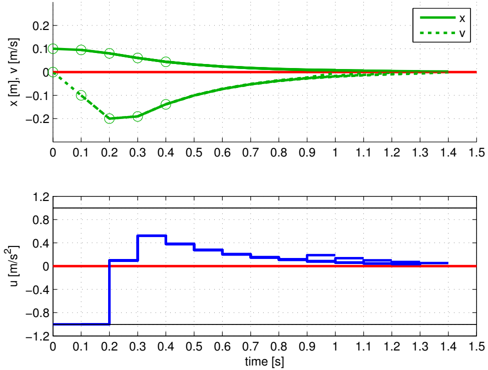
s.t. $s_{k+1} = f(s_k, u_k)$, $k = 0, \dots, N-1$, → **model** of the system evolution

$h(s_k, u_k) \leq 0$, $k = 0, \dots, N-1$, → **path constraints**

$s_0 = \hat{x}_0$. → **current state** of the system

Looking ahead in the future...

...design a control sequence minimizing the deviation from the reference



$$\min_{u,s} \quad \|s_N - s_{ref}\|_P^2 + \sum_{k=0}^{N-1} \|s_k - s_{ref}\|_Q^2 + \|u_k - u_{ref}\|_R^2 \quad \rightarrow \quad \textbf{deviation} \text{ from the reference}$$

$$\begin{aligned} \text{s.t.} \quad & s_{k+1} = f(s_k, u_k), & k = 0, \ldots, N-1, & \quad \rightarrow \quad \textbf{model} \text{ of the system evolution} \\ & h(s_k, u_k) \le 0, & k = 0, \ldots, N-1, & \quad \rightarrow \quad \textbf{path constraints} \\ & s_0 = \hat{x}_0. & & \quad \rightarrow \quad \textbf{current state} \text{ of the system} \end{aligned}$$

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

NLP cannot be solved instantaneously: better to apply

- suboptimal solution (almost) instantaneously?
- optimal solution after a longer time delay?

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$

$$h(s_k, u_k) \geq 0,$$

$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \quad \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} \quad \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + \quad \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$

$$\text{s.t.} \quad \Delta s_{k+1} = \quad + \quad \Delta s_k + \quad \Delta u_k,$$

$$+ \quad \Delta s_k + \quad \Delta u_k \geq 0,$$

$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \quad \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$
$$h(s_k, u_k) \geq 0,$$
$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \quad \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$
$$\text{s.t.} \quad \Delta s_{k+1} = \quad + \quad \Delta s_k + \quad \Delta u_k,$$
$$+ \quad \Delta s_k + \quad \Delta u_k \geq 0,$$
$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$
$$h(s_k, u_k) \geq 0,$$
$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} B \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$

$$\text{s.t.} \quad \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k,$$
$$h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0,$$
$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\begin{aligned}\min_{u,s}\quad & \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2 \\ \text{s.t.}\quad & s_{k+1} = f(s_k, u_k) \\ & h(s_k, u_k) \geq 0, \\ & s_0 = \hat{x}_i\end{aligned}$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\begin{aligned}\min_{\Delta u, \Delta s}\quad & \frac{1}{2}\begin{bmatrix}\Delta s & \Delta u\end{bmatrix} B \begin{bmatrix}\Delta s \\ \Delta u\end{bmatrix} + J^T \begin{bmatrix}\Delta s \\ \Delta u\end{bmatrix} \\ \text{s.t.}\quad & \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k, \\ & h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0, \\ & s_0 = \hat{x}_i\end{aligned}$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2^{nd} order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$

$$h(s_k, u_k) \geq 0,$$

$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} B \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$

$$\text{s.t.} \quad \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k,$$

$$h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0,$$

$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$
$$h(s_k, u_k) \geq 0,$$
$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} B \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$
$$\text{s.t.} \quad \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k,$$
$$h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0,$$
$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization: **ensure descent**, stepsize $\alpha \in (0, 1]$

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$
$$h(s_k, u_k) \geq 0,$$
$$s_0 = \hat{x}_i$$

Quadratic Problem Approximation

QP (for a given s, u)

$$\min_{\Delta u, \Delta s} \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} B \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$

$$\text{s.t.} \quad \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k,$$
$$h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0,$$
$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization: **ensure descent**, stepsize $\alpha \in (0, 1]$
6. Update $\begin{bmatrix} s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} s \\ u \end{bmatrix} + \alpha \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\begin{aligned}\min_{u,s}\quad & \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2 \\ \text{s.t.}\quad & s_{k+1} = f(s_k, u_k) \\ & h(s_k, u_k) \geq 0, \\ & s_0 = \hat{x}_i\end{aligned}$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization: **ensure descent**, stepsize $\alpha \in (0, 1]$
6. Update $\begin{bmatrix} s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} s \\ u \end{bmatrix} + \alpha \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$

Real Time Iterations

RTI at time i

$$\begin{aligned}\min_{\Delta u, \Delta s}\quad & \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} J^T J \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} \\ \text{s.t.}\quad & \Delta s_{k+1} = f + \frac{\partial f}{\partial s}\Delta s_k + \frac{\partial f}{\partial u}\Delta u_k \\ & h + \frac{\partial h}{\partial s}\Delta s_k + \frac{\partial h}{\partial u}\Delta u_k \geq 0, \\ & s_0 = \hat{x}_i\end{aligned}$$

Preparation Phase
Without knowing $\hat{x}_i$

- **Linearize**
- (Gauss-Newton $\Rightarrow B \succ 0$)
- Prepare the QP

Real-Time Dilemma and Real Time Iteration (RTI) Scheme

SQP for NMPC in a nutshell

NMPC at time i

$$\min_{u,s} \quad \sum_{k=0}^{N} \|s_k - x_{ref}\|_Q^2 + \sum_{k=0}^{N-1} \|u_k - u_{ref}\|_R^2$$

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$

$$h(s_k, u_k) \geq 0,$$

$$s_0 = \hat{x}_i$$

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u: need 2nd order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization: **ensure descent**, stepsize $\alpha \in (0, 1]$
6. Update $\begin{bmatrix} s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} s \\ u \end{bmatrix} + \alpha \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$

Real Time Iterations

RTI at time i

$$\min_{\Delta u, \Delta s} \quad \frac{1}{2} \begin{bmatrix} \Delta s & \Delta u \end{bmatrix} J^T J \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix} + J^T \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$$

$$\text{s.t.} \quad \Delta s_{k+1} = f + \frac{\partial f}{\partial s} \Delta s_k + \frac{\partial f}{\partial u} \Delta u_k$$

$$h + \frac{\partial h}{\partial s} \Delta s_k + \frac{\partial h}{\partial u} \Delta u_k \geq 0,$$

$$s_0 = \hat{x}_i$$

Preparation Phase
Without knowing $\hat{x}_i$

- **Linearize**
- (Gauss-Newton $\Rightarrow B \succ 0$)
- Prepare the QP

Feedback Phase:

- Solve QP once $\hat{x}_i$ available
  $\rightarrow$ same latency as linear MPC

CasADi

www.casadi.org

Key Properties of Casadi

- Open Source (LGPL)
- Automatic Differentiation
- Python interface

What is it good for?

- Easy and powerful way of formulating NLPs/OCPs
- Linked to NLP solvers
- Linked to Sundials integrators

ACADO Toolkit

www.acadotoolkit.org

ACADO Toolkit

`www.acadotoolkit.org`

Key Properties of ACADO Toolkit

- Open Source (LGPL)
- User friendly interfaces close to mathematical syntax
- Depends only on the C++ standard library

ACADO Toolkit

`www.acadotoolkit.org`

Key Properties of ACADO Toolkit

- Open Source (LGPL)
- User friendly interfaces close to mathematical syntax
- Depends only on the C++ standard library

Multiplatform support

- C++: Linux, OS X, Windows
- MATLAB

ACADO Toolkit

`www.acadotoolkit.org`

Key Properties of ACADO Toolkit

- Open Source (LGPL)
- User friendly interfaces close to mathematical syntax
- Depends only on the C++ standard library

Multiplatform support

- C++: Linux, OS X, Windows
- MATLAB

Fast implementations for real-time execution
$\longrightarrow$ ACADO Code Generation tool:

Export tailored solver in plain C code