Nonlinear Model Predictive Control

Mario Zanon, Alberto Bemporad

Thanks to
S. Gros, M. Vukov, J. Frasch, M. Diehl
for some of the material and their precious help

Course: Numerical Methods for Optimal Control

- Direct Nonlinear Optimal Control
- May 25-29, 2020
- 20 hours lectures
- 10 hours supervised assignments

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear feedback $u_k = -Ks_k$

$$s_{k+1} = (A - BK)s_k = A_K s_k$$

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear feedback $u_k = -Ks_k$

$$s_{k+1} = (A - BK)s_k = A_K s_k$$

stable if $$\max\left(|\lambda(A_K)|\right) \leq 1$$

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear feedback $u_k = -Ks_k$

$$s_{k+1} = (A - BK)s_k = A_K s_k$$

stable if $$\max\left(|\lambda(A_K)|\right) \leq 1$$

- How to choose K?

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear feedback $u_k = -Ks_k$

$$s_{k+1} = (A - BK)s_k = A_K s_k$$

stable if $\max\left(|\lambda(A_K)|\right) \leq 1$

- How to choose K?

What about **LQR**?

$$\begin{aligned}
\min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{\infty} \|s_k\|_Q^2 + \|u_k\|_R^2 \\
\text{s.t.} \quad & s_0 = \bar{x} \\
& s_{k+1} = As_k + Bu_k, \qquad k \geq 0 \\
& \lim_{k\to\infty} s_k = 0
\end{aligned}$$

- Linear system

$$s_{k+1} = As_k + Bu_k$$

- Linear feedback $u_k = -Ks_k$

$$s_{k+1} = (A - BK)s_k = A_K s_k$$

stable if $\max\left(|\lambda(A_K)|\right) \leq 1$

- How to choose K?

What about **LQR**?

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{\infty} \|s_k\|_Q^2 + \|u_k\|_R^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k \geq 0 \\ & \lim_{k\to\infty} s_k = 0 \end{aligned}$$

Equivalent to solving the **DARE** (discrete algebraic Riccati equation)

$$P = Q + A^T PA - A^T PBK$$

$$K = (R + B^T PB)^{-1} B^T PA$$

Note the equivalence

Horizon: ∞

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{\infty} \|s_k\|_Q^2 + \|u_k\|_R^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k \geq 0 \\ & \lim_{k\to\infty} s_k = 0 \end{aligned}$$

$\Leftrightarrow$

Horizon: N

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{N} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k = 0, \dots, N-1 \end{aligned}$$

with $N \geq 1$ and P from the DARE.

Note the equivalence

Horizon: ∞

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{\infty} \|s_k\|_Q^2 + \|u_k\|_R^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k \geq 0 \\ & \lim_{k\to\infty} s_k = 0 \end{aligned}$$

$\Leftrightarrow$

Horizon: N

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{N} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k = 0, \ldots, N-1 \end{aligned}$$

with $N \geq 1$ and P from the DARE.

The term $\frac{1}{2}\|s_N\|_P^2$ is called cost to go

Note the equivalence

Horizon: ∞

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{\infty} \|s_k\|_Q^2 + \|u_k\|_R^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k \geq 0 \\ & \lim_{k\to\infty} s_k = 0 \end{aligned}$$

$\Leftrightarrow$

Horizon: N

$$\begin{aligned} \min_{s,u} \quad & \frac{1}{2}\sum_{k=0}^{N} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \bar{x} \\ & s_{k+1} = As_k + Bu_k, \qquad k = 0, \dots, N-1 \end{aligned}$$

with $N \geq 1$ and P from the DARE.

The term $\frac{1}{2}\|s_N\|_P^2$ is called cost to go

If we don't want to solve the DARE

- Choose P large enough
- Solve the finite horizon problem: **Quadratic Program (QP)**

At each sampling instant i, solve the QP

$$\begin{aligned}\min_{u,s}\quad & \frac{1}{2}\sum_{k=0}^{N-1}\|s_k\|_Q^2+\|u_k\|_R^2+\frac{1}{2}\|s_N\|_P^2\\ \text{s.t.}\quad & s_0=\hat{x}_i\\ & s_{k+1}=A\,s_k+B\,u_k\end{aligned}$$

At each sampling instant i, solve the QP

$$\begin{aligned}\min_{u,s}\quad & \frac{1}{2}\sum_{k=0}^{N-1}\|s_k\|_Q^2+\|u_k\|_R^2+\frac{1}{2}\|s_N\|_P^2\\ \text{s.t.}\quad & s_0=\hat{x}_i\\ & s_{k+1}=A\,s_k+B\,u_k\end{aligned}\quad\Leftrightarrow\quad\begin{aligned}\min_{w}\quad & \frac{1}{2}w^TFw+f^Tw\\ \text{s.t.}\quad & Gw+g=0\end{aligned}$$

At each sampling instant i, solve the QP

$$\begin{aligned}\min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\, s_k + B\, u_k \end{aligned} \quad \Leftrightarrow \quad \begin{aligned}\min_{w} \quad & \frac{1}{2} w^T F w + f^T w \\ \text{s.t.} \quad & Gw + g = 0\end{aligned}$$

Lagrangian Function

$$\mathcal{L}(w,\lambda) = \frac{1}{2} w^T F w + f^T w - \lambda^T (Gw + g)$$

At each sampling instant i, solve the QP

$$\begin{aligned}\min_{u,s}\quad & \frac{1}{2}\sum_{k=0}^{N-1}\|s_k\|_Q^2+\|u_k\|_R^2+\frac{1}{2}\|s_N\|_P^2\\ \text{s.t.}\quad & s_0=\hat{x}_i\\ & s_{k+1}=A\,s_k+B\,u_k\end{aligned} \quad\Leftrightarrow\quad \begin{aligned}\min_{w}\quad & \frac{1}{2}w^TFw+f^Tw\\ \text{s.t.}\quad & Gw+g=0\end{aligned}$$

Lagrangian Function

$$\mathcal{L}(w,\lambda)=\frac{1}{2}w^TFw+f^Tw-\lambda^T(Gw+g)$$

First order necessary condition (FONC)

$$\nabla\mathcal{L}(w,\lambda)=0 \qquad\Rightarrow\qquad \begin{cases}Fw+f-G^T\lambda=0\\ Gw+g=0\end{cases}$$

At each sampling instant i, solve the QP

$$\begin{aligned} \min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\, s_k + B\, u_k \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{w} \quad & \frac{1}{2} w^T F w + f^T w \\ \text{s.t.} \quad & Gw + g = 0 \end{aligned}$$

Lagrangian Function

$$\mathcal{L}(w,\lambda) = \frac{1}{2} w^T F w + f^T w - \lambda^T (Gw + g)$$

First order necessary condition (FONC)

$$\nabla \mathcal{L}(w,\lambda) = 0 \quad \Rightarrow \quad \begin{cases} Fw + f - G^T\lambda = 0 \\ Gw + g = 0 \end{cases}$$

Solve a linear system:

$$\begin{bmatrix} F & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} w \\ -\lambda \end{bmatrix} = -\begin{bmatrix} f \\ g \end{bmatrix}$$

Treating Constrained Systems

$$\begin{aligned} \min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\, s_k + B\, u_k \end{aligned}$$

- LQR: unconstrained

Treating Constrained Systems

$$\begin{aligned}
\min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\
\text{s.t.} \quad & s_0 = \hat{x}_i \\
& s_{k+1} = A\, s_k + B\, u_k \\
& C\, s_k + D\, u_k + c \geq 0
\end{aligned}$$

- LQR: unconstrained
- MPC: state and input constraints

Treating Constrained Systems

$$\begin{aligned} \min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\,s_k + B\,u_k \\ & C\,s_k + D\,u_k + c \geq 0 \end{aligned}$$

- LQR: unconstrained
- MPC: state and input constraints
- $\|s_N\|_P^2$ only approximates the cost to go

Treating Constrained Systems

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

$$\text{s.t.} \quad s_0 = \hat{x}_i$$

$$s_{k+1} = A\, s_k + B\, u_k$$

$$C\, s_k + D\, u_k + c \geq 0$$

- LQR: unconstrained
- MPC: state and input constraints
- $\|s_N\|_P^2$ only approximates the cost to go

Handle explicitly:

- Actuator limitations, e.g. saturation of an input signal
- State constraints, e.g. concentration of a reactant
- Mixed state-input constraints

MPC yields a nonlinear control law!

At each sampling instant i, solve the QP

$$\begin{aligned} \min_{u,s} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\,s_k + B\,u_k \\ & C\,s_k + D\,u_k + c \geq 0 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{w} \quad & \frac{1}{2} w^T F w + f^T w \\ \text{s.t.} \quad & Gw + g = 0 \\ & Hw + h \geq 0 \end{aligned}$$

At each sampling instant i, solve the QP

$$\begin{aligned}\min_{u,s}\quad & \frac{1}{2}\sum_{k=0}^{N-1}\|s_k\|_Q^2+\|u_k\|_R^2+\frac{1}{2}\|s_N\|_P^2\\ \text{s.t.}\quad & s_0=\hat{x}_i\\ & s_{k+1}=A\,s_k+B\,u_k\\ & C\,s_k+D\,u_k+c\geq 0\end{aligned} \quad\Leftrightarrow\quad \begin{aligned}\min_{w}\quad & \frac{1}{2}w^TFw+f^Tw\\ \text{s.t.}\quad & Gw+g=0\\ & Hw+h\geq 0\end{aligned}$$

Lagrangian Function

$$\mathcal{L}(w,\lambda,\mu)=\frac{1}{2}w^TFw+f^Tw-\lambda^T(Gw+g)-\mu^T(Hw+h)$$

At each sampling instant i, solve the QP

$$\begin{aligned} \min_{u,s} \quad & \frac{1}{2} \sum_{k=0}^{N-1} \|s_k\|_Q^2 + \|u_k\|_R^2 + \frac{1}{2}\|s_N\|_P^2 \\ \text{s.t.} \quad & s_0 = \hat{x}_i \\ & s_{k+1} = A\, s_k + B\, u_k \\ & C\, s_k + D\, u_k + c \geq 0 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{w} \quad & \frac{1}{2} w^T F w + f^T w \\ \text{s.t.} \quad & Gw + g = 0 \\ & Hw + h \geq 0 \end{aligned}$$

Lagrangian Function

$$\mathcal{L}(w, \lambda, \mu) = \frac{1}{2} w^T F w + f^T w - \lambda^T (Gw + g) - \mu^T (Hw + h)$$

First order necessary condition (FONC): the KKT conditions

$$\nabla \mathcal{L}(w, \lambda, \mu) = 0 \quad \Rightarrow \quad \begin{cases} Fw + f - G^T \lambda - H^T \mu = 0 \\ Gw + g = 0 \\ Hw + h \geq 0 \\ \mu \geq 0 \\ \mu_i (Hw + h)_i = 0 \end{cases}$$

Solving the KKT conditions

$$Fw + f - G^T\lambda - H^T\mu = 0$$
$$Gw + g = 0$$
$$Hw + h \geq 0$$
$$\mu \geq 0$$
$$\mu_i(Hw + h)_i = 0$$

Solving the KKT conditions

$$Fw + f - G^T\lambda - H^T\mu = 0$$
$$Gw + g = 0$$
$$Hw + h \geq 0$$
$$\mu \geq 0$$
$$\mu_i (Hw + h)_i = 0$$

The Active Set method

Let $\mathbb{A}$ be the set of **active** constraints

$$Fw + f - G^T\lambda - H^T\mu = 0$$
$$Gw + g = 0$$
$$H_{\mathbb{A}} w + h_{\mathbb{A}} = 0$$
$$\mu_{\bar{\mathbb{A}}} = 0$$

- Guess $\mathbb{A}$
- Solve the AS-KKT system
- Update $\mathbb{A}$

Solving the KKT conditions

$$
\begin{aligned}
&Fw + f - G^T\lambda - H^T\mu = 0 \\
&Gw + g = 0 \\
&Hw + h \geq 0 \\
&\mu \geq 0 \\
&\mu_i (Hw + h)_i = 0
\end{aligned}
$$

The Active Set method

Let $\mathbb{A}$ be the set of **active** constraints

$$
\begin{aligned}
&Fw + f - G^T\lambda - H^T\mu = 0 \\
&Gw + g = 0 \\
&H_{\mathbb{A}} w + h_{\mathbb{A}} = 0 \\
&\mu_{\bar{\mathbb{A}}} = 0
\end{aligned}
$$

- Guess $\mathbb{A}$
- Solve the AS-KKT system
- Update $\mathbb{A}$

The Interior Point method

$$
\begin{aligned}
&Fw + f - G^T\lambda - H^T\mu = 0 \\
&Gw + g = 0 \\
&Hw + h + s = 0 \\
&\mu_i s_i = \tau \\
&\mu \geq 0 \\
&s \geq 0
\end{aligned}
$$

- Choose τ "big"
- Solve the IP-KKT system
- Perform linesearch
- update τ

QP solvers for MPC

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**
- Inequalities: **interior point** or **active set** method

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**
- Inequalities: **interior point** or **active set** method

Nonconvex QP: **NP-hard** problem

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**
- Inequalities: **interior point** or **active set** method

Nonconvex QP: **NP-hard** problem

Classes of QP solvers:

- Active-set
- Interior-point
- First-order methods (difficult to use for nonconvex problems)

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**
- Inequalities: **interior point** or **active set** method

Nonconvex QP: **NP-hard** problem

Classes of QP solvers:

- Active-set
- Interior-point
- First-order methods (difficult to use for nonconvex problems)

Many **reliable QP solvers** available:

- qpOASES, qpDUNES
- FORCES, HPMPC / HPIPM
- ODYSQP
- many others

QP solvers for MPC

Convex QP:

- No inequalities: solve a **linear system**
- Inequalities: **interior point** or **active set** method

Nonconvex QP: **NP-hard** problem

Classes of QP solvers:

- Active-set
- Interior-point
- First-order methods (difficult to use for nonconvex problems)

Many **reliable QP solvers** available:

- qpOASES, qpDUNES
- FORCES, HPMPC / HPIPM
- ODYSQP
- many others

Condensing

- Eliminate states (cost N^2)
- Solve dense QP

Sparse linear algebra

- Exploit the qp structure

Linear system?

Linear MPC at time i

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

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

$$C\, s_k + D\, u_k \geq 0,$$

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

Linear system?

Linear MPC at time i

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

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

$$C\, s_k + D\, u_k \geq 0,$$

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

1. Linear dynamics
2. Linear path constraints
3. Solve a QP at each iteration
4. Extremely fast for small to medium scale problems

Linear system?

Linear MPC at time i

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

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

$$C\, s_k + D\, u_k \geq 0,$$

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

1. Linear dynamics
2. Linear path constraints
3. Solve a QP at each iteration
4. Extremely fast for small to medium scale problems

Nonlinear system?

- Linearize at x_{ref}, u_{ref}, use linear MPC

Linear system?

Linear MPC at time i

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

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

$$C\, s_k + D\, u_k \geq 0,$$

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

1. Linear dynamics
2. Linear path constraints
3. Solve a QP at each iteration
4. Extremely fast for small to medium scale problems

Nonlinear system?

- Linearize at x_{ref}, u_{ref}, use linear MPC
- or...

Nonlinear MPC at time i

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

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

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

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

Linear system?

Linear MPC at time i

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

$$\text{s.t.} \quad s_{k+1} = A\, s_k + B\, u_k$$
$$C\, s_k + D\, u_k \geq 0,$$
$$s_0 = \hat{x}_i$$

1. Linear dynamics
2. Linear path constraints
3. Solve a QP at each iteration
4. Extremely fast for small to medium scale problems

Nonlinear system?

- Linearize at x_{ref}, u_{ref}, use linear MPC
- or...

Nonlinear MPC at time i

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

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

Problem is non-convex, use NLP solver

SQP for NMPC in a nutshell

NMPC at time i

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

Iterative procedure (at each time i):

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)

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

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

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

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

Iterative procedure (at each time i):

1. Given current guess s, u

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)

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

Iterative procedure (at each time i):

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

SQP for NMPC in a nutshell

Quadratic Problem Approximation

NMPC at time i

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

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

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

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

QP (for a given s, u)

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

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

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

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

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

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 2^{nd} order derivatives for B
3. Make sure Hessian $B \succ 0$: avoid negative curvature
4. Solve QP

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)

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

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

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

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

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 (e.g. line-search): **ensure descent**, stepsize $\alpha \in (0, 1]$

SQP for NMPC in a nutshell

Quadratic Problem Approximation

NMPC at time i

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

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

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

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

QP (for a given s, u)

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

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

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

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

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 (e.g. line-search): **ensure descent**, stepsize $\alpha \in (0, 1]$
6. Update $\begin{bmatrix} s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} s \\ u \end{bmatrix} + \alpha \begin{bmatrix} \Delta s \\ \Delta u \end{bmatrix}$ and iterate

Linear system

Continuous time:

$$\dot{x}(t) = A_\mathrm{c} x(t) + B_\mathrm{c} u(t)$$

Discrete time:

$$s_{k+1} = A\, s_k + B\, u_k$$

Discretization over a time interval $t \in [t_k, t_{k+1}]$, input $u(t) = u_k$

$$A = e^{A_\mathrm{c}\left(t_{k+1} - t_k\right)},$$

$$B = \int_{t_k}^{t_{k+1}} e^{A_\mathrm{c}\tau} B_\mathrm{c} \mathrm{d}\tau$$

Linear system	Nonlinear system

Continuous time:

$$\dot{x}(t) = A_{\mathrm{c}} x(t) + B_{\mathrm{c}} u(t) \qquad\qquad \dot{x}(t) = f_{\mathrm{c}}\left(x(t), u(t)\right)$$

Discrete time:

$$s_{k+1} = A\, s_k + B\, u_k \qquad\qquad s_{k+1} = f\left(s_k, u_k\right)$$

Discretization over a time interval $t \in [t_k, t_{k+1}]$**, input** $u(t) = u_k$

$$A = e^{A_{\mathrm{c}}\left(t_{k+1} - t_k\right)},$$

$$B = \int_{t_k}^{t_{k+1}} e^{A_{\mathrm{c}}\tau} B_{\mathrm{c}} \mathrm{d}\tau$$

Integration of function f_{c} can be **complex**, possibly **iterative implicit** (algorithm) !!

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Integration (with sensitivities)

Consider $\dot{x} = f_c(x, u)$

Discretize with explicit Euler:

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$x_1 = h f_{\mathrm{c}}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$x_1 = h f_{\mathrm{c}}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = h f_{\mathrm{c}}(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Integration (with sensitivities)

Consider $\dot{x} = f_c(x, u)$

Discretize with explicit Euler:

$$x_1 = hf_c(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = hf_c(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Sensitivities wrt states:

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$x_1 = hf_{\mathrm{c}}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = hf_{\mathrm{c}}(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Sensitivities wrt states:

$$A_1 = \frac{\partial x_1}{\partial x_0} = (I + h\nabla_x f_{\mathrm{c}}(x_0, u))$$

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$
\begin{aligned}
x_1 &= h f_{\mathrm{c}}(x_0, u) + x_0, & A_1 &= \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, & B_1 &= \frac{\mathrm{d}x_1}{\mathrm{d}u} \\
x_2 &= h f_{\mathrm{c}}(x_1, u) + x_1, & A_2 &= \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, & B_2 &= \frac{\mathrm{d}x_2}{\mathrm{d}u}
\end{aligned}
$$

Sensitivities wrt states:

$$
\begin{aligned}
A_1 &= \frac{\partial x_1}{\partial x_0} = (I + h \nabla_x f_{\mathrm{c}}(x_0, u)) \\
A_2 &= \frac{\partial x_2}{\partial x_1} \frac{\partial x_1}{\partial x_0} = (I + h \nabla_x f_{\mathrm{c}}(x_1, u))\, A_1
\end{aligned}
$$

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$x_1 = hf_{\mathrm{c}}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = hf_{\mathrm{c}}(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Sensitivities wrt states:

$$A_1 = \frac{\partial x_1}{\partial x_0} = (I + h\nabla_x f_{\mathrm{c}}(x_0, u))$$

$$A_2 = \frac{\partial x_2}{\partial x_1}\frac{\partial x_1}{\partial x_0} = (I + h\nabla_x f_{\mathrm{c}}(x_1, u))\, A_1$$

For the controls it's a bit trickier:

Integration (with sensitivities)

Consider $\dot{x} = f_\mathrm{c}(x, u)$

Discretize with explicit Euler:

$$x_1 = hf_\mathrm{c}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = hf_\mathrm{c}(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Sensitivities wrt states:

$$A_1 = \frac{\partial x_1}{\partial x_0} = (I + h\nabla_x f_\mathrm{c}(x_0, u))$$

$$A_2 = \frac{\partial x_2}{\partial x_1}\frac{\partial x_1}{\partial x_0} = (I + h\nabla_x f_\mathrm{c}(x_1, u))\, A_1$$

For the controls it's a bit trickier:

$$B_1 = \frac{\partial x_1}{\partial u} = h\nabla_u f_\mathrm{c}(x_0, u)$$

Integration (with sensitivities)

Consider $\dot{x} = f_{\mathrm{c}}(x, u)$

Discretize with explicit Euler:

$$x_1 = hf_{\mathrm{c}}(x_0, u) + x_0, \qquad A_1 = \frac{\mathrm{d}x_1}{\mathrm{d}x_0}, \qquad B_1 = \frac{\mathrm{d}x_1}{\mathrm{d}u}$$

$$x_2 = hf_{\mathrm{c}}(x_1, u) + x_1, \qquad A_2 = \frac{\mathrm{d}x_2}{\mathrm{d}x_0}, \qquad B_2 = \frac{\mathrm{d}x_2}{\mathrm{d}u}$$

Sensitivities wrt states:

$$A_1 = \frac{\partial x_1}{\partial x_0} = \left(I + h\nabla_x f_{\mathrm{c}}(x_0, u)\right)$$

$$A_2 = \frac{\partial x_2}{\partial x_1}\frac{\partial x_1}{\partial x_0} = \left(I + h\nabla_x f_{\mathrm{c}}(x_1, u)\right) A_1$$

For the controls it's a bit trickier:

$$B_1 = \frac{\partial x_1}{\partial u} = h\nabla_u f_{\mathrm{c}}(x_0, u)$$

$$B_2 = \frac{\partial x_2}{\partial u} + \frac{\partial x_2}{\partial x_1}\frac{\partial x_1}{\partial u} = h\nabla_u f_{\mathrm{c}}(x_1, u) + \left(I + h\nabla f_{\mathrm{c}}(x_1, u)\right) B_1$$

Integration (with sensitivities)

There are many numerical schemes:

Integration (with sensitivities)

There are many numerical schemes:

- Explicit Euler is usually not the most efficient method! Inaccuracy: $O(h)$

Integration (with sensitivities)

There are many numerical schemes:

- Explicit Euler is usually not the most efficient method! Inaccuracy: $O(h)$
- Explicit Runge-Kutta of order 4 is rather successful. Inaccuracy: $O(h^4)$

$$k_1 = f_\mathrm{c}(x, u) \qquad k_2 = f_\mathrm{c}\left(x + \frac{h}{2}k_1, u\right)$$

$$k_3 = f_\mathrm{c}\left(x + \frac{h}{2}k_2, u\right) \qquad k_4 = f_\mathrm{c}(x + hk_3, u)$$

$$x^+ = x + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Integration (with sensitivities)

There are many numerical schemes:

- Explicit Euler is usually not the most efficient method! Inaccuracy: $O(h)$
- Explicit Runge-Kutta of order 4 is rather successful. Inaccuracy: $O(h^4)$

$$k_1 = f_c(x, u) \qquad k_2 = f_c\left(x + \frac{h}{2}k_1, u\right)$$

$$k_3 = f_c\left(x + \frac{h}{2}k_2, u\right) \qquad k_4 = f_c(x + hk_3, u)$$

$$x^+ = x + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

- Implicit schemes have desirable properties (stiff systems)
  Simplest example (implicit Euler): $x^+ = x + hf_c(x^+, u)$

Integration (with sensitivities)

There are many numerical schemes:

- Explicit Euler is usually not the most efficient method! Inaccuracy: $O(h)$
- Explicit Runge-Kutta of order 4 is rather successful. Inaccuracy: $O(h^4)$

$$k_1 = f_\mathrm{c}(x, u) \qquad k_2 = f_\mathrm{c}\left(x + \frac{h}{2}k_1, u\right)$$

$$k_3 = f_\mathrm{c}\left(x + \frac{h}{2}k_2, u\right) \qquad k_4 = f_\mathrm{c}(x + hk_3, u)$$

$$x^+ = x + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

- Implicit schemes have desirable properties (stiff systems)
  Simplest example (implicit Euler): $x^+ = x + hf_\mathrm{c}(x^+, u)$
- Collocation = Implicit Runge-Kutta

Integration (with sensitivities)

There are many numerical schemes:

- Explicit Euler is usually not the most efficient method! Inaccuracy: $O(h)$
- Explicit Runge-Kutta of order 4 is rather successful. Inaccuracy: $O(h^4)$

$$k_1 = f_\mathrm{c}(x, u) \qquad k_2 = f_\mathrm{c}\left(x + \frac{h}{2}k_1, u\right)$$

$$k_3 = f_\mathrm{c}\left(x + \frac{h}{2}k_2, u\right) \qquad k_4 = f_\mathrm{c}(x + hk_3, u)$$

$$x^+ = x + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

- Implicit schemes have desirable properties (stiff systems)
  Simplest example (implicit Euler): $x^+ = x + hf_\mathrm{c}(x^+, u)$
- Collocation = Implicit Runge-Kutta
- Exponential integrators, e.g.

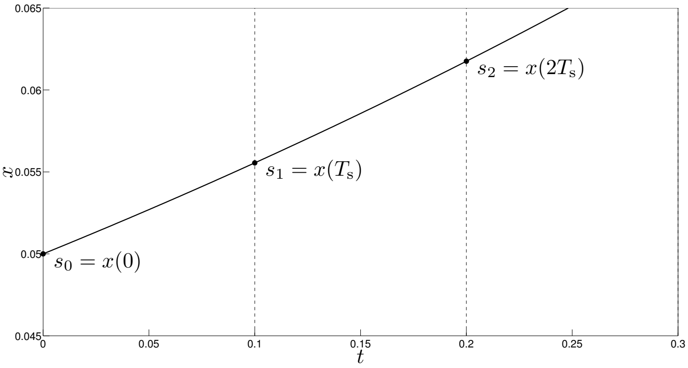
$$x^+ = Ax + Bu, \quad A = e^{h\nabla_x f_\mathrm{c}(x,u)}, \quad B = \int_0^h e^{\tau\nabla_x f_\mathrm{c}(x,u)}\nabla_u f_\mathrm{c}(x, u)\mathrm{d}\tau$$

How to Discretize the System?

How to Discretize the System?

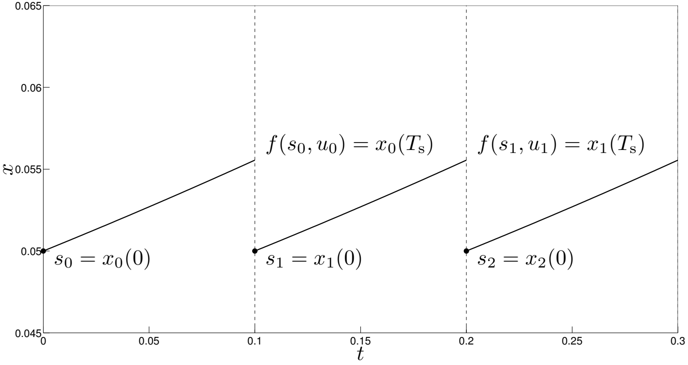
- Single Shooting:

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



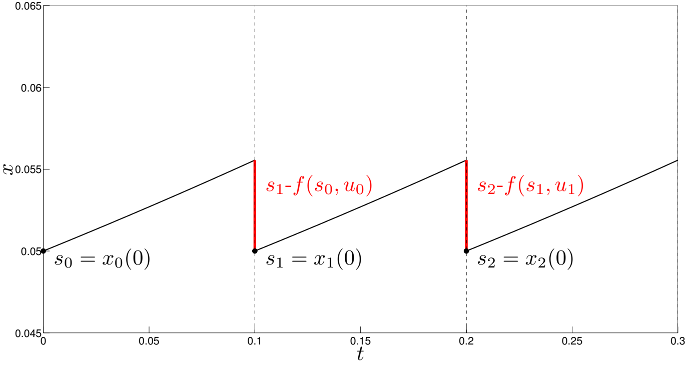
How to Discretize the System?

- 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**



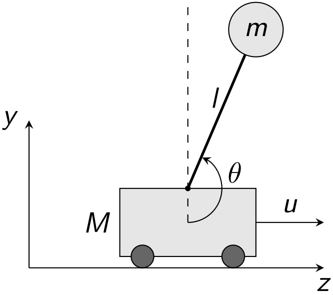
How to Discretize the System?

- 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**

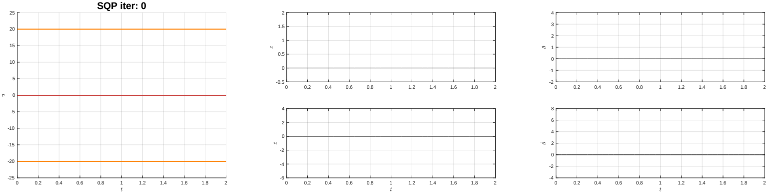


Single Shooting

$$\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}$$

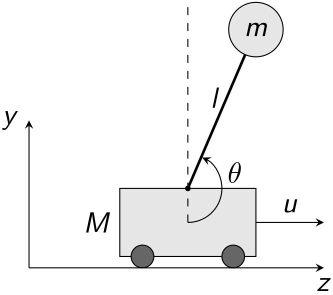


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

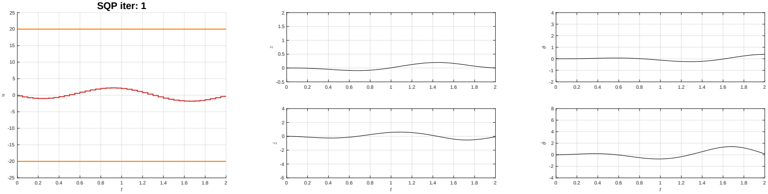


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad && t \in [0,2] \\
& -20 \le u(t) \le 20, && t \in [0,2]
\end{aligned}
$$

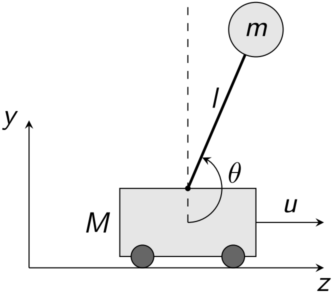


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

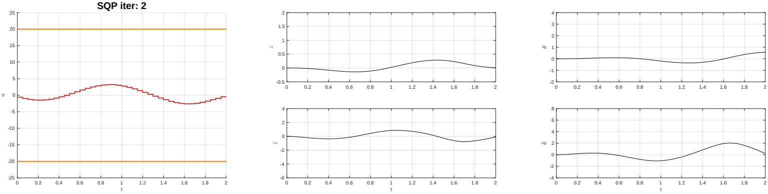


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \le u(t) \le 20, \qquad t \in [0, 2]
\end{aligned}
$$

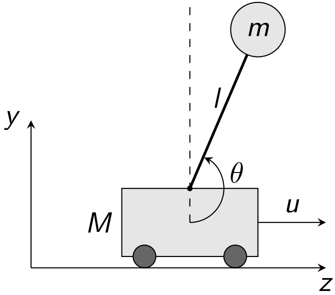


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

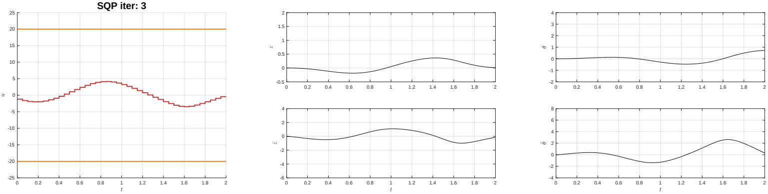


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

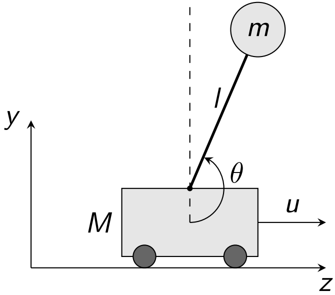


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

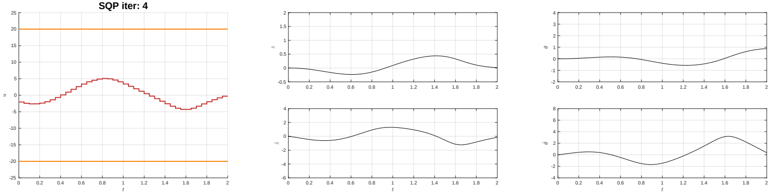


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

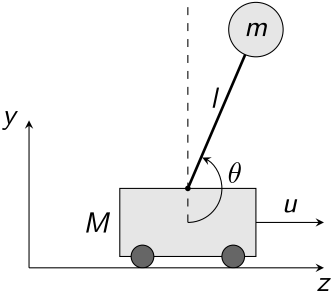


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

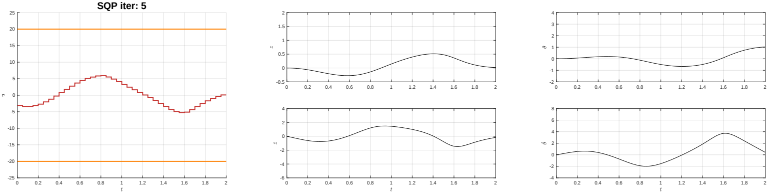


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

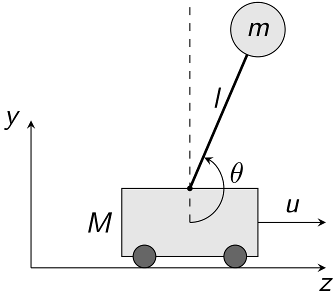


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

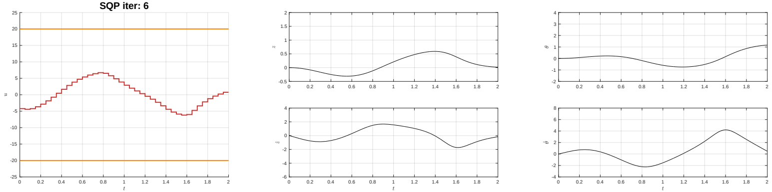


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

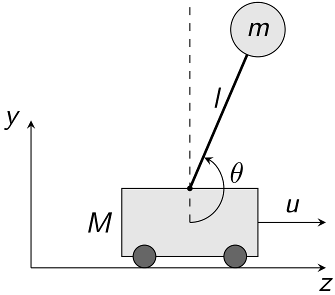


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

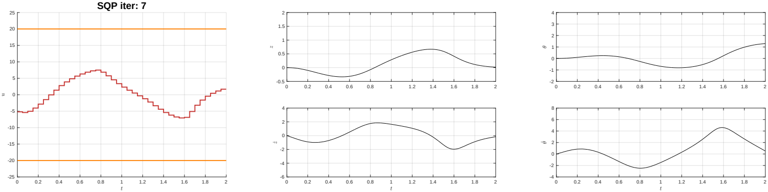


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

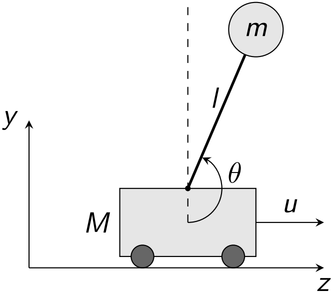


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

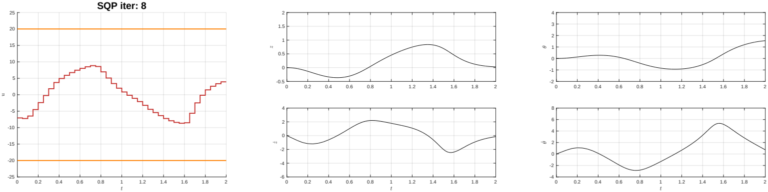


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$



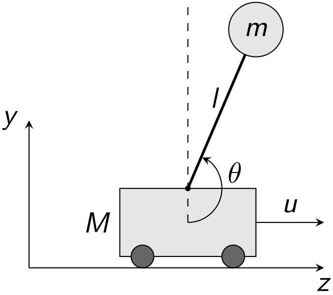
$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$



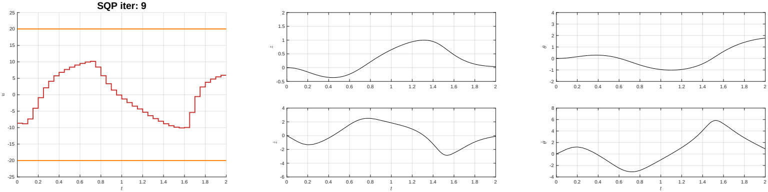
Single Shooting

$$\min_{u(\cdot)} \quad \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}$$

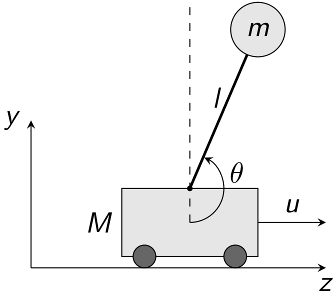


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

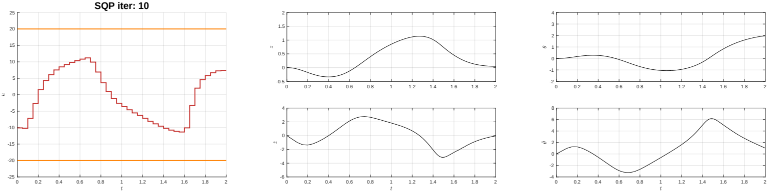


Single Shooting

$$\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2] \\
& -20 \le u(t) \le 20, \qquad t \in [0,2]
\end{aligned}$$

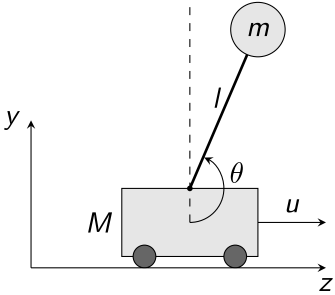


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

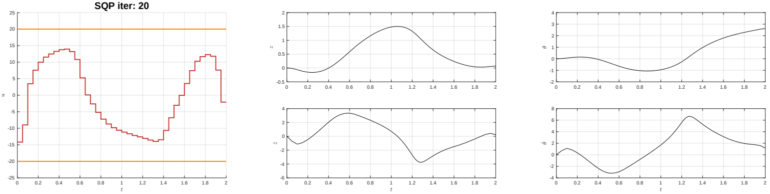


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

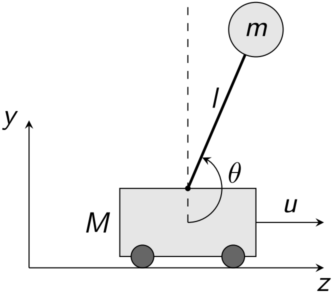


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

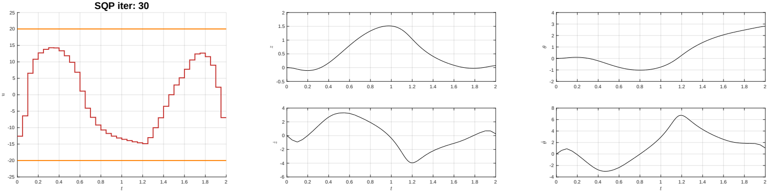


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

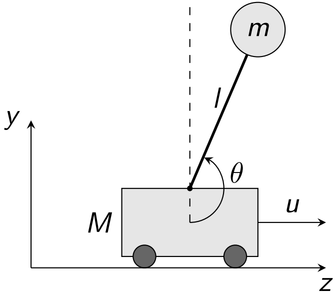


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

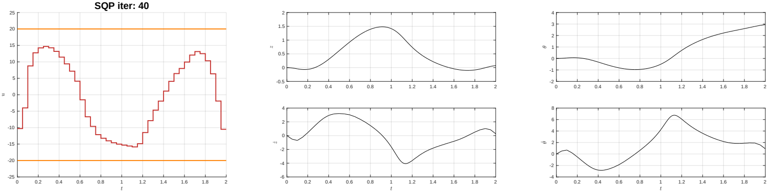


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

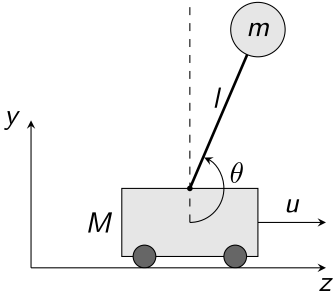


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

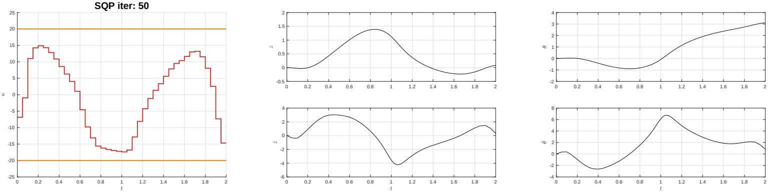


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0,2]
\end{aligned}
$$

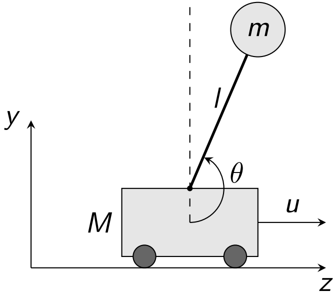


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

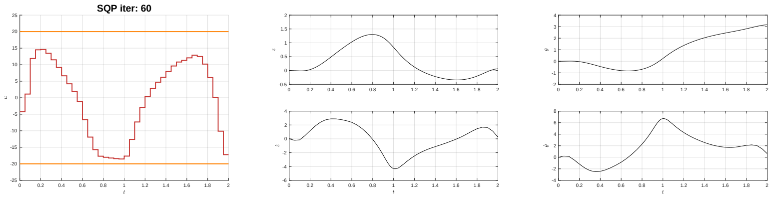


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \le u(t) \le 20, \qquad t \in [0, 2]
\end{aligned}
$$

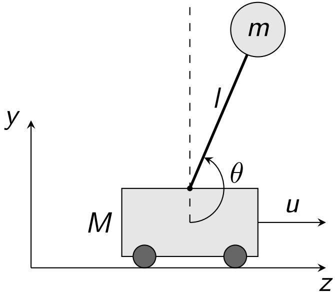


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

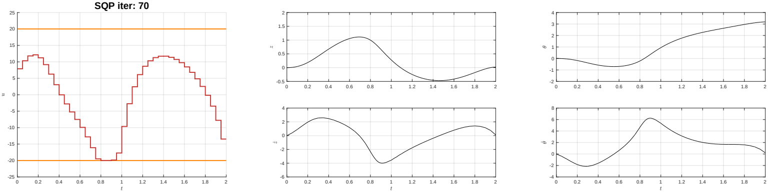


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

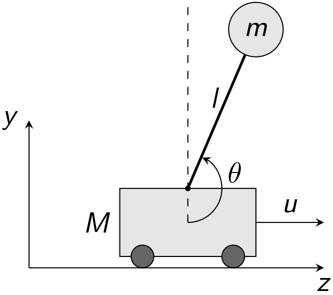


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

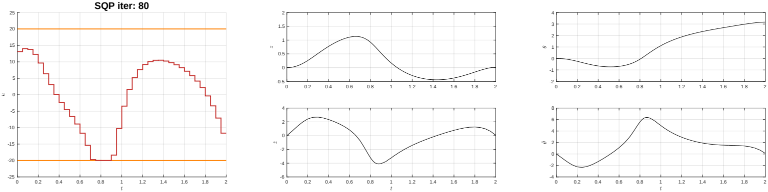


Single Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

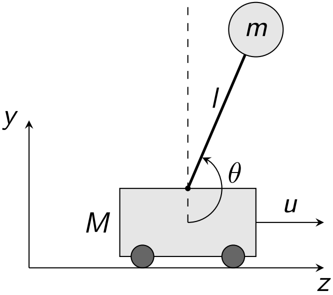


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

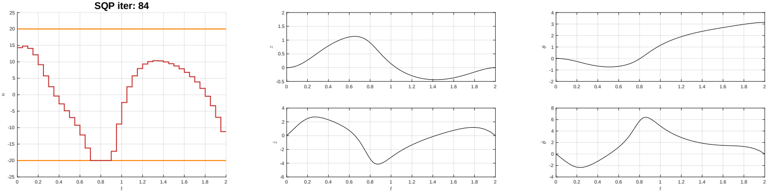


Single Shooting

$$\begin{aligned}\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\ \text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\ & x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\ & \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2] \\ & -20 \leq u(t) \leq 20, \qquad t \in [0,2]\end{aligned}$$

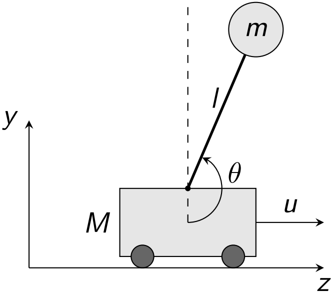


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

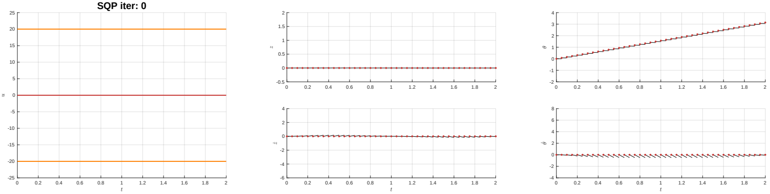


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \,\mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$



$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$



Multiple Shooting

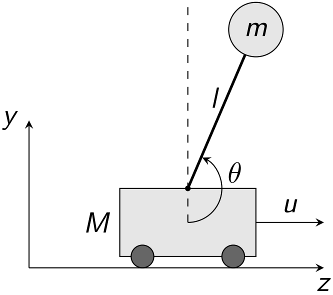
$$\min_{u(\cdot)} \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t$$

$$\text{s.t.} \quad x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$$

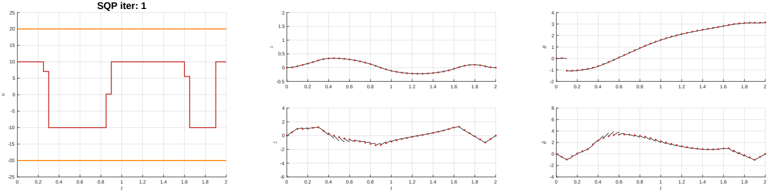
$$x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top$$

$$\dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2]$$

$$-20 \leq u(t) \leq 20, \qquad t \in [0, 2]$$

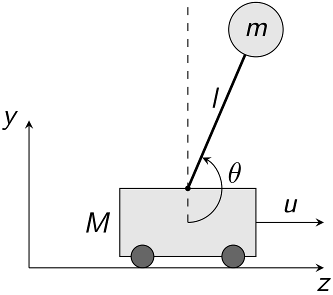


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

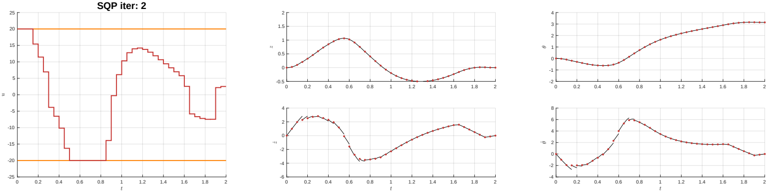


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

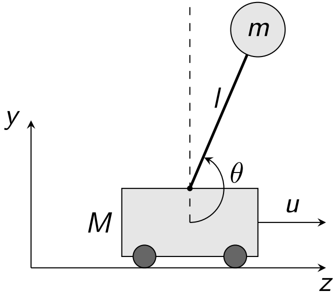


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

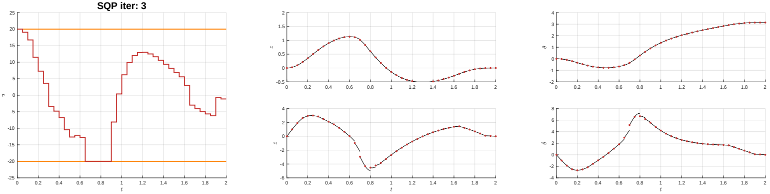


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

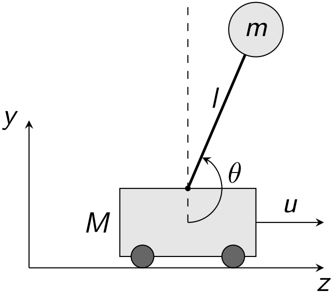


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

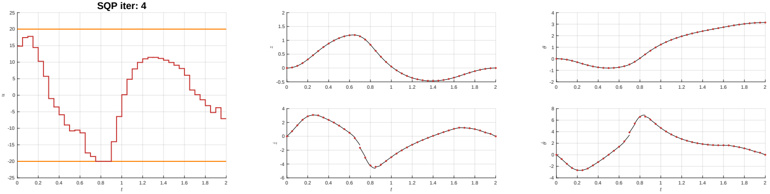


Multiple Shooting

$$\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}$$



$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta) \sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta) \sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$



Multiple Shooting

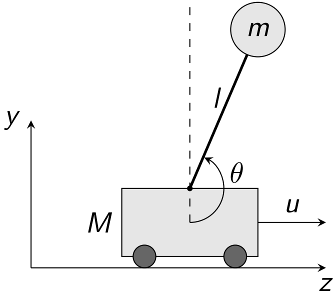
$$\min_{u(\cdot)} \quad \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t$$

$$\text{s.t.} \quad x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$$

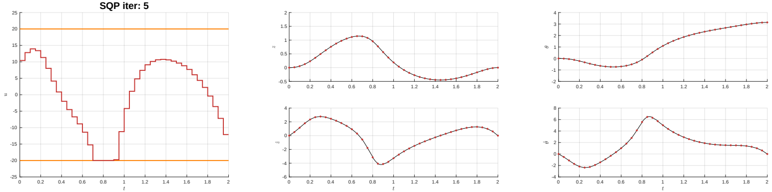
$$x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top$$

$$\dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2]$$

$$-20 \le u(t) \le 20, \qquad t \in [0,2]$$

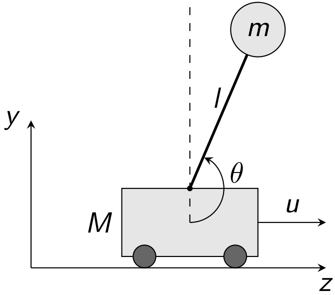


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

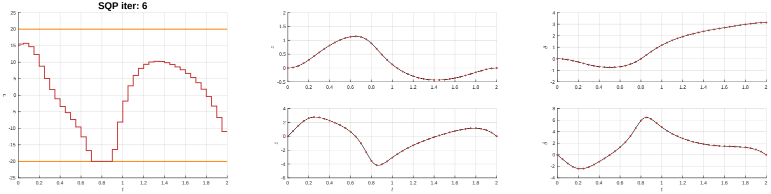


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \leq u(t) \leq 20, \qquad t \in [0, 2]
\end{aligned}
$$

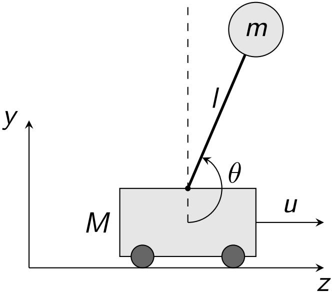


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

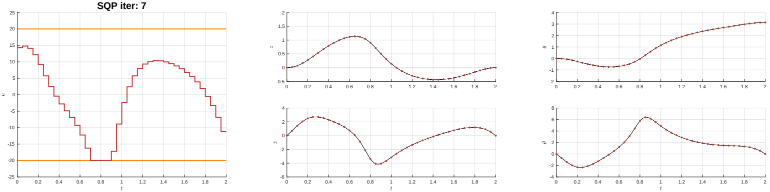


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0, 2] \\
& -20 \le u(t) \le 20, \qquad t \in [0, 2]
\end{aligned}
$$

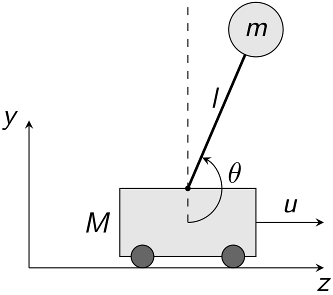


$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml \sin(\theta)\omega^2 + mg \cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml \cos(\theta)\sin(\theta)\omega^2 + (M+m)g \sin(\theta) + u \cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$

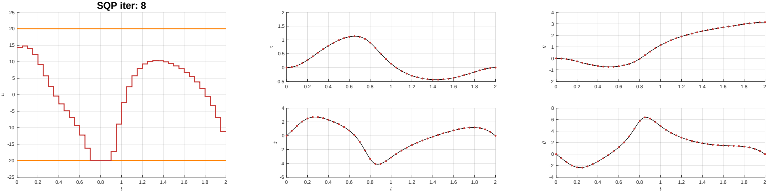


Multiple Shooting

$$\begin{aligned}\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\ \text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\ & x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\ & \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2] \\ & -20 \leq u(t) \leq 20, \qquad t \in [0,2]\end{aligned}$$

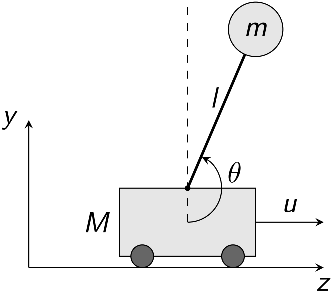


$$\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\frac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}$$

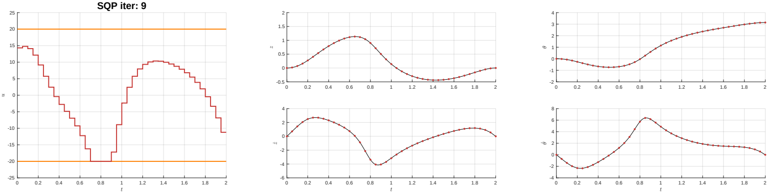


Multiple Shooting

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_{t_0}^{2} u(t)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \\
& x(2) = \begin{bmatrix} 0 & \pi & 0 & 0 \end{bmatrix}^\top \\
& \dot{x}(t) = F(x(t), u(t)), \qquad t \in [0,2] \\
& -20 \le u(t) \le 20, \qquad t \in [0,2]
\end{aligned}
$$



$$
\begin{bmatrix} \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \dfrac{ml\sin(\theta)\omega^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2} \\ -\dfrac{(ml\cos(\theta)\sin(\theta)\omega^2 + (M+m)g\sin(\theta) + u\cos(\theta))}{L(M + m - m(\cos(\theta))^2)} \end{bmatrix}
$$



Multiple Shooting vs Single Shooting

- **Better**: unstable systems

Multiple Shooting vs Single Shooting

- **Better**: **unstable systems**
- **Better**: **initialization of states** at intermediate nodes

Multiple Shooting vs Single Shooting

- **Better**: **unstable systems**
- **Better**: **initialization of states** at intermediate nodes
- **Warning**: leads to **bigger QP/NLP**
  - S. Shooting: $n_x + (N-1)n_u$ opt. vars $(x_0, u_0, u_1, \ldots, u_{N-1})$
  - M. Shooting: $Nn_x + (N-1)n_u$ opt. vars $(x_0, u_0, x_1, u_1, \ldots, x_N)$

Multiple Shooting vs Single Shooting

- **Better**: **unstable systems**
- **Better**: **initialization of states** at intermediate nodes
- **Warning**: leads to **bigger QP/NLP**
  - S. Shooting: $n_x + (N-1)n_u$ opt. vars $(x_0, u_0, u_1, \ldots, u_{N-1})$
  - M. Shooting: $Nn_x + (N-1)n_u$ opt. vars $(x_0, u_0, x_1, u_1, \ldots, x_N)$
- **Good news!**: **after integration**, all $x_k,\ k = 1, \ldots, N$ can be eliminated

  → **Condensing**: reduce to the size of Single Shooting (using the dynamic constraints)

  → **Efficient Sparse Linear Algebra** can be very effective, especially for long horizons

Multiple Shooting vs Single Shooting

- **Better**: **unstable systems**
- **Better**: **initialization of states** at intermediate nodes
- **Warning**: leads to **bigger QP/NLP**
  - S. Shooting: $n_x + (N-1)n_u$ opt. vars $(x_0, u_0, u_1, \ldots, u_{N-1})$
  - M. Shooting: $Nn_x + (N-1)n_u$ opt. vars $(x_0, u_0, x_1, u_1, \ldots, x_N)$
- **Good news!**: **after integration**, all x_k, $k = 1, \ldots, N$ can be eliminated
  - → **Condensing**: reduce to the size of Single Shooting (using the dynamic constraints)
  - → **Efficient Sparse Linear Algebra** can be very effective, especially for long horizons
- **Continuity conditions**:
  - S. Shooting: imposed by the integration
  - M. Shooting: imposed by the QP/NLP

Let's get a closer look at SQP

Let's get a closer look at SQP

QP (for a given s, u)

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

Linearize

- f: evaluate integrator
- $\frac{\partial f}{\partial s}$, $\frac{\partial f}{\partial u}$: differentiate integrator
- h: evaluate nonlinear function
- $\frac{\partial h}{\partial s}$, $\frac{\partial h}{\partial u}$: differentiate nonlinear function
- $B = \text{diag}(Q, \ldots, Q, R \ldots, R) + \left\langle \lambda, \frac{\partial^2 f}{\partial w^2} \right\rangle + \left\langle \mu, \frac{\partial^2 h}{\partial w^2} \right\rangle, \quad w = \begin{bmatrix} s \\ u \end{bmatrix}$
- $J = 2\, w^T \text{diag}(Q, \ldots, Q, R \ldots, R)$

Let's get a closer look at SQP

QP (for a given s, u)

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

Ensure $B \succ 0$

- Exact Hessian: add curvature to the negative directions
  Quadratic convergence

Let's get a closer look at SQP

QP (for a given s, u)

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

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

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

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

Ensure $B \succ 0$

- Exact Hessian: add curvature to the negative directions
  Quadratic convergence
- BFGS update: $B_{k+1} = B_k + \frac{B_k \sigma \sigma^T B_k}{\sigma^T B_k \sigma} + \frac{\gamma \gamma^T}{\sigma^T \gamma}$
  Superlinear convergence

Let's get a closer look at SQP

QP (for a given s, u)

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

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

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

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

Ensure $B \succ 0$

- Exact Hessian: add curvature to the negative directions
  Quadratic convergence
- BFGS update: $B_{k+1} = B_k + \frac{B_k \sigma \sigma^T B_k}{\sigma^T B_k \sigma} + \frac{\gamma \gamma^T}{\sigma^T \gamma}$
  Superlinear convergence
- **Gauss-Newton** approximation: $B \approx J^T J$ (for linear MPC it is exact!)
  Linear convergence

Let's get a closer look at SQP

QP (for a given s, u)

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

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

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

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

Iterate to convergence

- All previous steps are repeated until convergence!
- Computations can become very long
- Cannot apply the control instantaneously

Can we exploit the MPC structure to be faster?

What about:

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step $\rightarrow$ no need to iterate

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step $\rightarrow$ no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint $\rightarrow$ faster convergence, clever computations

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step → no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint → faster convergence, clever computations
- No globalization → need to enforce $s_0 = \hat{x}_i$

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step → no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint → faster convergence, clever computations
- No globalization → need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation → only 1^{st} order derivatives, Hessian $B \succ 0$

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step → no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint → faster convergence, clever computations
- No globalization → need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation → only 1^{st} order derivatives, Hessian $B \succ 0$

Result:

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step $\rightarrow$ no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint $\rightarrow$ faster convergence, clever computations
- No globalization $\rightarrow$ need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation $\rightarrow$ only 1^{st} order derivatives, Hessian $B \succ 0$

Result:

- Converge while the system evolves
  next SQP iteration takes place on the new problem $\hat{x}_{i+1}$

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step → no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint → faster convergence, clever computations
- No globalization → need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation → only 1^{st} order derivatives, Hessian $B \succ 0$

Result:

- Converge while the system evolves
  next SQP iteration takes place on the new problem $\hat{x}_{i+1}$
- Need to have a good initial guess
  better to shift (to be continued...)

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step → no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint → faster convergence, clever computations
- No globalization → need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation → only 1^{st} order derivatives, Hessian $B \succ 0$

Result:

- Converge while the system evolves
  next SQP iteration takes place on the new problem $\hat{x}_{i+1}$
- Need to have a good initial guess
  better to shift (to be continued...)
- We are essentially doing path-following
  with a generalized tangential predictor

Can we exploit the MPC structure to be faster?

What about:

- 1 Newton step $\rightarrow$ no need to iterate
- Initial value embedding: $s_0 = \hat{x}_i$ as a constraint $\rightarrow$ faster convergence, clever computations
- No globalization $\rightarrow$ need to enforce $s_0 = \hat{x}_i$
- Gauss-Newton Hessian approximation $\rightarrow$ only 1$^{\text{st}}$ order derivatives, Hessian $B \succ 0$

Result:

- Converge while the system evolves
  next SQP iteration takes place on the new problem $\hat{x}_{i+1}$
- Need to have a good initial guess
  better to shift (to be continued...)
- We are essentially doing path-following
  with a generalized tangential predictor

Under some (mild) conditions, the SQP solution is closely tracked

Standard SQP

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
3. Make sure Hessian $B \succ 0$
4. Solve QP
5. Globalization (e.g. line-search)
6. Update and iterate

Standard SQP

NMPC at time i

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

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

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

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

Real Time Iterations

RTI at time i

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

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

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

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

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u
3. Make sure Hessian $B \succ 0$
4. Solve QP
5. Globalization (e.g. line-search)
6. Update and iterate

Standard SQP

NMPC at time i

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

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

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

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

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u
3. Make sure Hessian $B \succ 0$
4. Solve QP
5. Globalization (e.g. line-search)
6. Update and iterate

Real Time Iterations

RTI at time i

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

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

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

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

Preparation Phase

Without knowing $\hat{x}_i$

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

Standard SQP

NMPC at time i

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

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

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

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

Iterative procedure (at each time i):

1. Given current guess s, u
2. **Linearize** at s, u
3. Make sure Hessian $B \succ 0$
4. Solve QP
5. Globalization (e.g. line-search)
6. Update and iterate

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

Linear MPC at time i

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

RTI at time i

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

Linear MPC at time i

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

At each time i:

1. Solve the QP

RTI at time i

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

At each time i:

1. Solve the QP

Linear MPC at time i

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

At each time i:

1. Solve the QP
2. Wait

RTI at time i

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

At each time i:

1. Solve the QP
2. Compute the new linearization of the constraints
3. Prepare the new QP

Linear MPC at time i

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

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

$$C_k s_k + D_k u_k \geq 0,$$

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

At each time i:

1. Solve the QP
2. Wait

RTI at time i

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

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

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

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

At each time i:

1. Solve the QP
2. Compute the new linearization of the constraints
3. Prepare the new QP

RTI differs from linear MPC in the sense that the constraints are re-linearized at each time instant on the current trajectory rather than only once on the reference trajectory

Embedded Solver

RTI at time i

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

Properties:

- Fixed problem dimensions
- Specific structure

Embedded Solver

RTI at time i

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

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

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

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

Properties:

- Fixed problem dimensions
- Specific structure

Tailored C code

- Exploit the structure and minimize number of operations
- No dynamic memory allocation
- Minimize branching in the exported code
- Optimized linear algebra routines

Embedded Solver

RTI at time i

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

Properties:

- Fixed problem dimensions
- Specific structure

Tailored C code

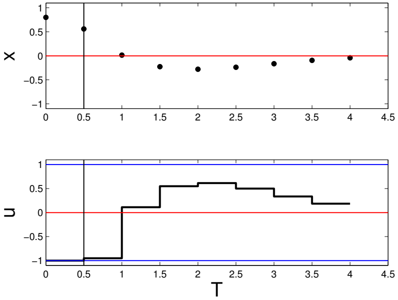
- Exploit the structure and minimize number of operations
- No dynamic memory allocation
- Minimize branching in the exported code
- Optimized linear algebra routines

ACADO / ACADOS

- Multiple shooting
- Real time iterations

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $\rightarrow \|[s,u]-[s^*,u^*]\| = o(e^2_{\mathrm{guess}})$
- **Guess**: shift the solution at the previous step

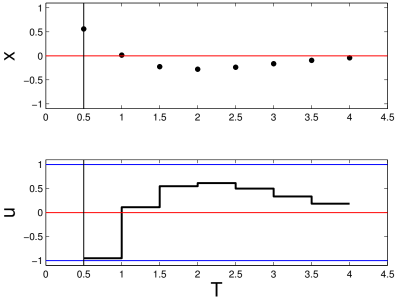


Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $\rightarrow \|[s,u]-[s^*,u^*]\| = o(e^2_{\mathrm{guess}})$
- **Guess**: shift the solution at the previous step

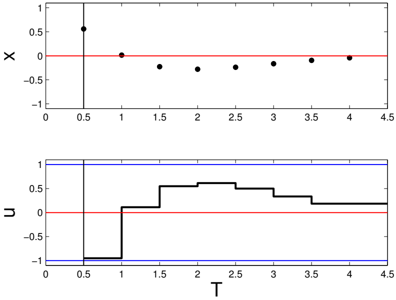


Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $\rightarrow \|[s, u] - [s^*, u^*]\| = o(e^2_{\text{guess}})$
- **Guess**: shift the solution at the previous step

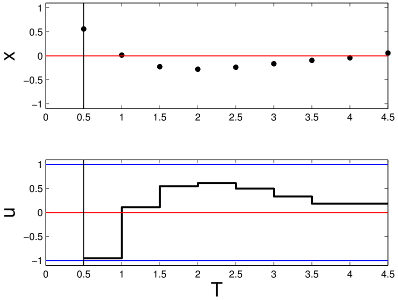


Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $\rightarrow \|[s,u]-[s^*,u^*]\| = o(e^2_{\text{guess}})$
- **Guess**: shift the solution at the previous step

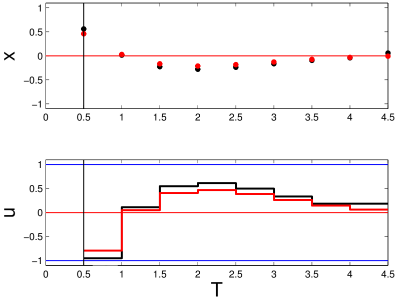


Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $$\rightarrow \|[s,u]-[s^*,u^*]\| = o(e_{\text{guess}}^2)$$
- **Guess**: shift the solution at the previous step

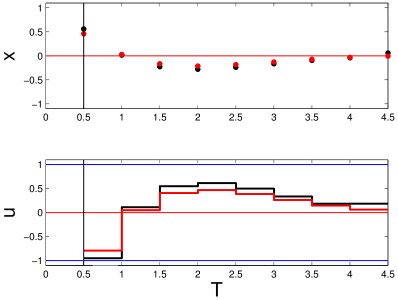


Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Importance of the Initial Guess

- **RTI** (single Newton step): 1^{st} order correction
  $\rightarrow \|[s,u]-[s^*,u^*]\| = o(e^2_{\text{guess}})$
- **Guess**: shift the solution at the previous step



Guess error e_{guess} small if...

1. $\{u^{i-1}, s^{i-1}\}$ close to $\{u^{i-1^*}, s^{i-1^*}\}$
2. s_0^i close to $\hat{x}^i$

Key for algorithmic reliability

1. Sample fast enough
2. Warm start

Tuning: Trial-and-error, experience-based

Tuning: Trial-and-error, experience-based

- Positive-definite cost
  - not necessary, but recommended
  - helps convergence

Tuning: Trial-and-error, experience-based

- Positive-definite cost
  - not necessary, but recommended
  - helps convergence
- For pure setpoint tracking
  1. use a scaled identity $W_{i,i} = \sigma_i^{-2}$, where $\sigma_i =$ range of state i
  2. simulate and adjust the weights as $W_{i,i} = \omega_i \sigma_i^{-2}$
  3. iterate until performance is satisfactory

Tuning: Trial-and-error, experience-based

- Positive-definite cost
  - not necessary, but recommended
  - helps convergence
- For pure setpoint tracking
  1. use a scaled identity $W_{i,i} = \sigma_i^{-2}$, where σ_i = range of state i
  2. simulate and adjust the weights as $W_{i,i} = \omega_i \sigma_i^{-2}$
  3. iterate until performance is satisfactory
- If you have a local linear controller you wish to imitate
  - controller matching for feedback K [Zanon,Bemporad, w.i.p.]

$$\begin{aligned} \min_{\alpha,\beta,W,P} \quad & \gamma\beta - \alpha \\ \text{s.t.} \quad & P = Q + A^\top PA - (S^\top + A^\top PB)K \\ & (R + A^\top PB)K = S + B^\top PA \\ & \beta I \succ W \succ \alpha I \end{aligned} \qquad \text{where} \quad W = \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix}$$

Tuning: Trial-and-error, experience-based

- Positive-definite cost
  - not necessary, but recommended
  - helps convergence
- For pure setpoint tracking
  1. use a scaled identity $W_{i,i} = \sigma_i^{-2}$, where σ_i = range of state i
  2. simulate and adjust the weights as $W_{i,i} = \omega_i \sigma_i^{-2}$
  3. iterate until performance is satisfactory
- If you have a local linear controller you wish to imitate
  - controller matching for feedback K [Zanon,Bemporad, w.i.p.]

$$\begin{aligned} \min_{\alpha,\beta,W,P} \quad & \gamma\beta - \alpha \\ \text{s.t.} \quad & P = Q + A^\top PA - (S^\top + A^\top PB)K \\ & (R + A^\top PB)K = S + B^\top PA \\ & \beta I \succ W \succ \alpha I \end{aligned} \qquad \text{where} \quad W = \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix}$$

- If there is a clear "economic" objective
  - automatic tuning [Zanon, Gros, Diehl, JPC2016]

Handling Infeasibility

Handling Infeasibility

Path constraints might become infeasible

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

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible
- limited controllability of $h(s_k, u_k)$ at low k

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible
- limited controllability of $h(s_k, u_k)$ at low k

Slack Variables Reformulation

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

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible
- limited controllability of $h(s_k, u_k)$ at low k

Slack Variables Reformulation

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

- No effect for $h(s_k, u_k) \geq 0$

Handling Infeasibility

Path constraints might become infeasible

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

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible
- limited controllability of $h(s_k, u_k)$ at low k

Slack Variables Reformulation

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

- No effect for $h(s_k, u_k) \geq 0$
- Strong penalty for $h(s_k, u_k) \leq 0$

  $\rightarrow$ choose W large enough

Handling Infeasibility

Path constraints might become infeasible

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

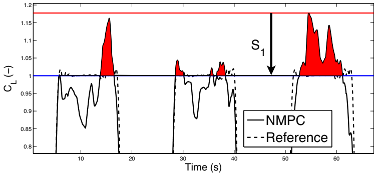
$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k)$$
$$h(s_k, u_k) \geq 0,$$
$$s_0 = \hat{x}_i$$

- $\hat{x}_i$ imposed by the (perturbed) system
- perturbations might make $h(s_k, u_k) \geq 0$ infeasible
- limited controllability of $h(s_k, u_k)$ at low k

Slack Variables Reformulation

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

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



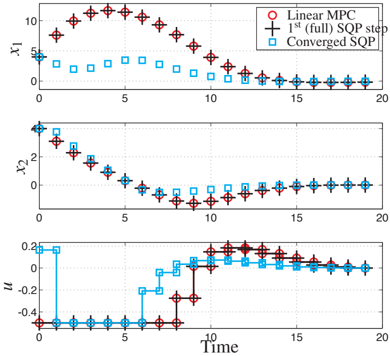
RTI Illustration

Consider a simple (discrete-time) problem

$$\begin{aligned}
\min_{u,x} \quad & \sum_{k=0}^{N} \|x_k\|^2 + 20 \sum_{k=0}^{N-1} \|u_k\|^2 \\
\text{s.t.} \quad & x_0 = \hat{x}_i, \\
& x_{k+1} = 0.9 x_k + \begin{bmatrix} \sin\left(\begin{bmatrix} 0 & 1 \end{bmatrix} x_k\right) \\ u_k + u_k^3 \end{bmatrix}, \\
& |u_k| < 0.5, \quad k = 0, \ldots, N-1,
\end{aligned}$$

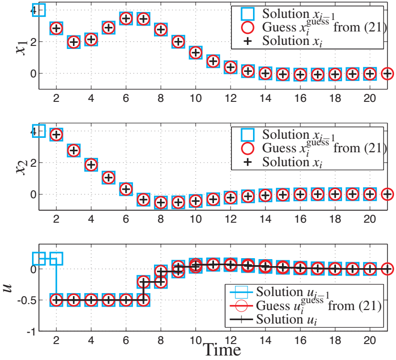
Consider a simple (discrete-time) problem

Initialise everything at the reference



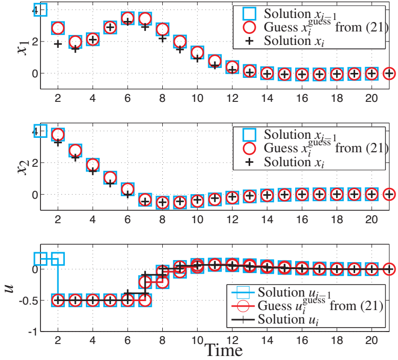
Consider a simple (discrete-time) problem

Shift from previous solution, no noise

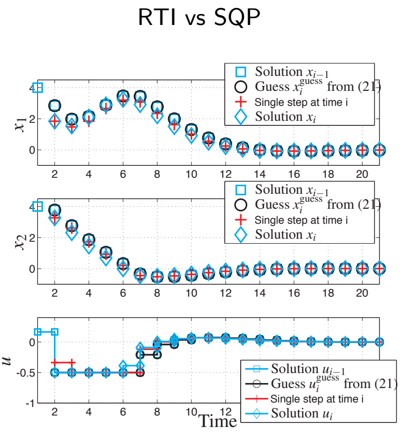


Consider a simple (discrete-time) problem

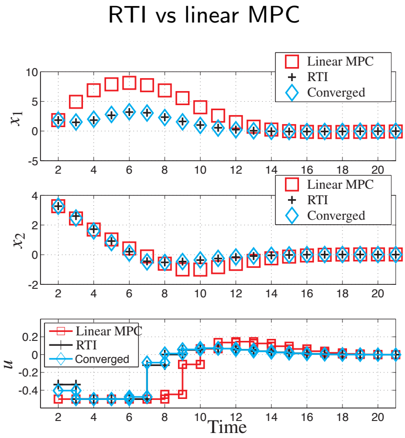
Shift from previous solution, process noise



Consider a simple (discrete-time) problem

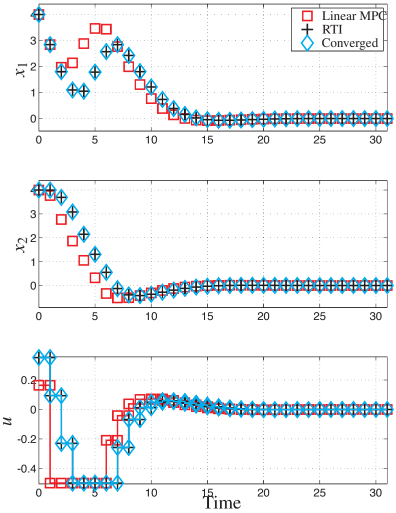


Consider a simple (discrete-time) problem



Consider a simple (discrete-time) problem

Closed-loop: RTI, linear MPC and SQP, no noise



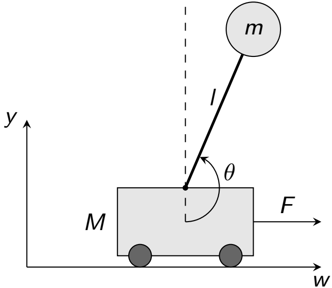
Consider a simple (continuous-time) problem

Pendulum on a cart:

$$\ddot{w} = \frac{ml\sin(\theta)\dot{\theta}^2 + mg\cos(\theta)\sin(\theta) + u}{M + m - m(\cos(\theta))^2},$$

$$\ddot{\theta} = -\frac{ml\cos(\theta)\sin(\theta)\dot{\theta}^2 + u\cos(\theta) + (M+m)g\sin(\theta)}{l(M + m - m(\cos(\theta))^2)},$$

with $M = 1\ \mathrm{kg}$, $m = 0.1\ \mathrm{kg}$, $l = 0.5\ \mathrm{m}$, $g = 9.81\ \mathrm{m/s^2}$.

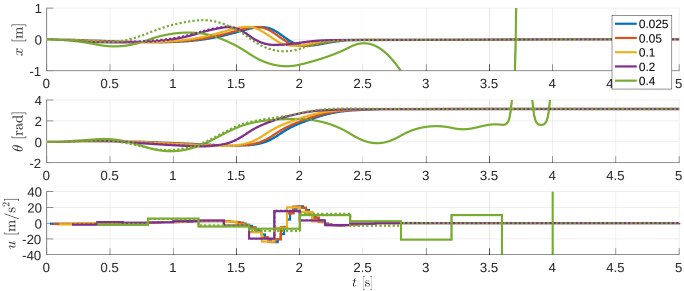


- Prediction horizon: 2 s
- Stage cost matrices:

$$Q = \mathrm{diag}(\begin{bmatrix} 10 & 10 & 0.1 & 0.1 \end{bmatrix}),$$
$$R = 0.01$$

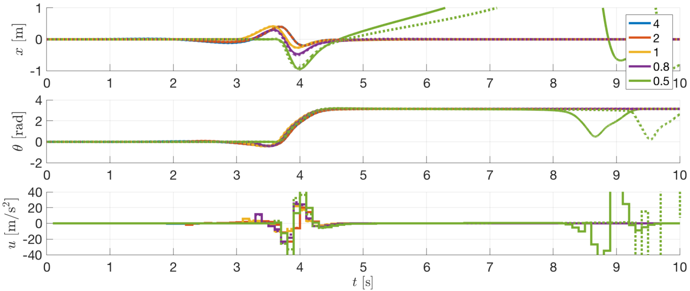
Consider a simple (continuous-time) problem

Sampling time (s)



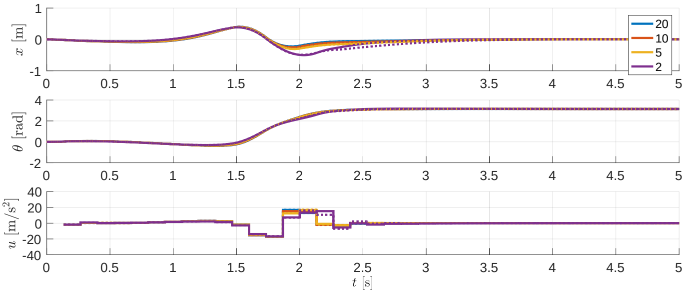
Consider a simple (continuous-time) problem

Prediction horizon (s)



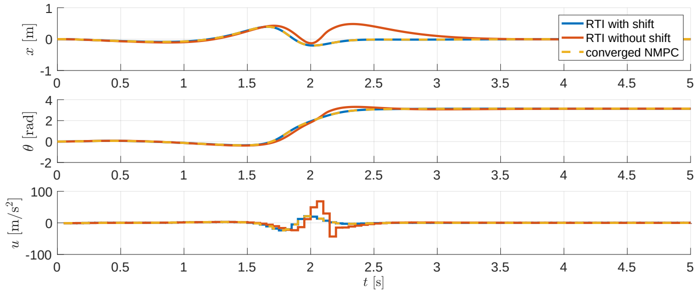
Consider a simple (continuous-time) problem

Integrator accuracy (steps of explicit Euler)

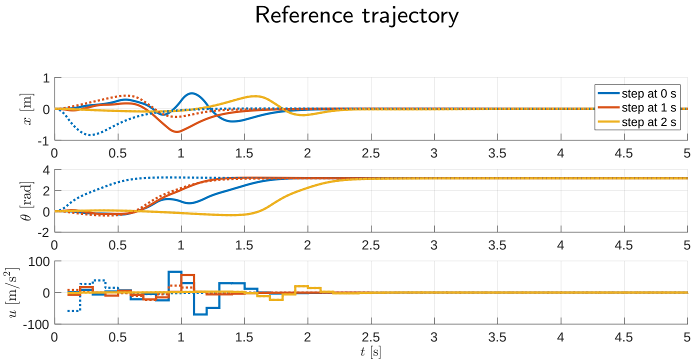


Consider a simple (continuous-time) problem

Shift



Consider a simple (continuous-time) problem



Thank you for your attention!