MPC Exercise 2 - Discretization

We want to discretize the optimization problem

$$
\begin{aligned}
\underset{\bar{u}(\cdot;t)}{\text{minimize}} \quad & \int_{t}^{t+T} \|\bar{x}(\tau;t)\|_Q^2 + \|\bar{u}(\tau;t)\|_R^2 \,\mathrm{d}\tau + \|\bar{x}(t+T;t)\|_P^2 \\
\text{subject to} \quad & \dot{\bar{x}} = f(\bar{x},\bar{u}) \\
& \bar{x}(t;t) = x(t) \\
& \bar{u}(\tau;t) \in \mathcal{U} \\
& \bar{x}(t+T;t) \in \Omega_\alpha.
\end{aligned}
$$

MPC Exercise 2 - Discretization

We want to discretize the optimization problem
$\rightsquigarrow$ Using RK4, we obtain (compare Exercise 1)

$$
\begin{aligned}
\underset{\bar{u}(\cdot;t)}{\text{minimize}} \quad & \sum_{k=t}^{t+N-1} \left(\delta \|\bar{x}(k|t)\|_Q^2 + \delta \|\bar{u}(k|t)\|_R^2 \right) + \|\bar{x}(t+N|t)\|_P^2 \\
\text{subject to} \quad & \bar{x}(k+1|t) = \bar{x}(k|t) + \frac{\delta}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\
& \bar{x}(t|t) = x(t) \\
& \bar{u}(k|t) \in \mathcal{U} \\
& \bar{x}(t+N|t) \in \Omega_\alpha .
\end{aligned}
$$

Institut für Regelungstechnik
Leibniz Universität Hannover

Problem 2: Design a terminal region, controller and cost:

Design Procedure

1. Compute K such that Jacobi linearization is stabilized

MPC Exercise 2 - Design Procedure

Problem 2: Design a terminal region, controller and cost:

Design Procedure

1. Compute K such that Jacobi linearization is stabilized
2. Choose κ satisfying $\kappa < -\max \Re\{\lambda(A_K)\}$ and solve

$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q^*$$

to obtain P

MPC Exercise 2 - Design Procedure

Leibniz Universität Hannover

Problem 2: Design a terminal region, controller and cost:

Design Procedure

1. Compute K such that Jacobi linearization is stabilized
2. Choose κ satisfying $\kappa < -\max \Re\{\lambda(A_K)\}$ and solve

$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q^*$$

   to obtain P
3. Find largest possible α_1 such that $Kx \in \mathcal{U}$ for all $x \in \mathcal{X}^f_{\alpha_1}$

Institut für Regelungstechnik
Leibniz Universität Hannover

MPC Exercise 2 - Design Procedure

Problem 2: Design a terminal region, controller and cost:

Design Procedure

1. Compute K such that Jacobi linearization is stabilized
2. Choose κ satisfying $\kappa < -\max \Re\{\lambda(A_K)\}$ and solve
$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q^*$$
to obtain P
3. Find largest possible α_1 such that $Kx \in \mathcal{U}$ for all $x \in \mathcal{X}^f_{\alpha_1}$
4. Find largest possible $\alpha \in (0, \alpha_1]$ such that $L_\phi \leq \frac{\kappa \lambda_{\min}(P)}{\|P\|}$ holds

MPC Exercise 2 - Design Procedure

Problem 2: Design a terminal region, controller and cost:

Alternative (less conservative) for Step 4):

Solve optimization problem

$$\begin{aligned} \max_x \ & x^T P\phi(x) - \kappa x^T P x \\ \text{s.t. } & x^T P x \leq \alpha \end{aligned}$$

Leibniz Universität Hannover

Problem 2: Design a terminal region, controller and cost:

Alternative (less conservative) for Step 4):

Solve optimization problem

$$\max_x \; x^T P\phi(x) - \kappa x^T P x$$
$$\text{s.t.} \; x^T P x \leq \alpha$$

Iterate this by reducing α from α_1 until optimal value of is nonpositive

MPC Exercise 2 - `fmincon`

Leibniz Universität Hannover

- `xsol = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)` solves

$$
\begin{aligned}
\min_{x} \quad & f(x) \\
\text{s.t.} \quad & c(x) \leq 0, \qquad c_{eq}(x) = 0, \\
& Ax \leq b, \qquad A_{eq}x = b_{eq}, \\
& lb \leq x \leq ub \text{ (element-wise)}
\end{aligned}
$$

(`x0`: Initial guess for optimization variable `x`)

⇒ $f(x)$, $c(x)$, and $c_{eq}(x)$ have to be defined (in Matlab) as functions:

```
function cost = fun(x,param1,param2)
```
(`x`: optimization variable, `param`: fixed parameter(s) in $f(x)$)
```
  cost = ???
```
(Implementation of $f(x)$ in Matlab)
```
end
```
and
```
param1 = ...
```
(Some fixed value)
```
param2 = ...
```
(Some fixed value)
```
xsol = fmincon(@(x) fun(x,param), ...)
```

Institut für Regelungstechnik
Leibniz Universität Hannover

MPC Exercise 2 - `fmincon`

- `xsol = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)` solves

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & c(x) \leq 0, \qquad c_{eq}(x) = 0, \\ & Ax \leq b, \qquad A_{eq}x = b_{eq}, \\ & lb \leq x \leq ub \text{ (element-wise)} \end{aligned}$$

(`x0`: Initial guess for optimization variable `x`)

⇒ $f(x)$, $c(x)$, and $c_{eq}(x)$ have to be defined (in Matlab) as functions:

`function [c ceq] = nonlcon(x,param)`

(`x`: optimization variable, `param`: fixed parameter(s) in $c(x)$, $c_{eq}(x)$)

`c = ???` (Implementation of $c(x)$, $c_{eq}(x)$ in Matlab)

`ceq = ???` (`c / ceq = []`, if there are no nonlinear (in)equality constraints)

`end`

and

`param = ...` (Some fixed value)

`xsol = fmincon(...,@(x) nonlcon(x,param))`