Learning-Based Predictive Control
Dr. L. Hewing
Prof. M. Zeilinger
Prof. L. Fagiano

EECI 2023

Exercise 2
Robust Learning-Based MPC

# 1 Pen & Paper

## 1.1 Set-Membership Estimation for Linear Parametric State Space Models

Consider a linear state space system

$$x(k+1) = A_\theta x(k) + B_\theta u(k) + w(k)$$

with $\mathcal{W} = \{w \mid H_w w \leq h_w\}$ and

$$[A_\theta,\ B_\theta] = [A,\ B] + \sum_{i=1}^{n_\theta} [A_{\theta_i},\ B_{\theta_i}]\, \theta_i$$

where $\theta = [\theta_1,\, \theta_2]^T \in \Omega = \{\theta \mid H_\theta \theta \leq h_\theta\}$. Given a new state measurement $x(k+1)$,

1. Compute the set $\Delta := \{\theta \mid \exists w \in \mathcal{W}, x(k+1) = A_\theta x(k) + B_\theta u(k) + w\}$ as $\{\theta \mid H_\Delta \theta \leq h_\Delta\}$
2. Compute the update non-falsified parameter set $\Omega^+ = \Omega \cap \Delta$ as $\{\theta \mid H_{\Omega^+} \theta \leq h_{\Omega^+}\}$

# 2 Programming Exercise

## Robust Learning-based MPC of Mass-Spring-Damper

### Preliminaries

You are provided with the following coding template files:

| | | |
|---|---|---|
| `E02_RobustLBMPC.mlx` | Main (live) script | (to be edited throughout) |
| `SetMembership.m` | SetMembership Estimtation Class | (to be edited in Section 2) |
| `PLMPC.m` | Robust Performance Learning MPC Class | (to be edited in Section 3) |
| `SMMPC.m` | Set Membership MPC Class | (to be edited in Section 4) |
| `RobustMPC.m` | Robust MPC Class (Solution from E01) | |
| `RobustMPC.mat` | Robust MPC instance (Solution from E01) | |
| `LinearSystem.m` | Helper Class | |
| `SimulateMSD.m` | Helper Function | |

Areas in the code to be modified are indicated by

```
% ----- Start Modifying Code Here ------
%
% The existing (incomplete) code in the modifiable areas serves as a hint
%
% ----- End Modifying Code Here ------
```

### Problem Description

In this exercise we investigate performance learning and a set-membership (adaptive) robust linear MPC for the simple mass-spring damper system of Exercise 1. Different to the first exercise, we now consider a mismatch between the true system

$$x(k+1) = A_{\text{true}}x(k) + B_{\text{true}}u(k) + w(k), \qquad A_{\text{true}} = \begin{bmatrix} 1 & T_s \\ -k_{\text{true}}T_s & 1 - d_{\text{true}}T_s \end{bmatrix}, \qquad B_{\text{true}} = \begin{bmatrix} 0 \\ T_s \end{bmatrix}$$

and the nominal model used in the MPC

$$x_{i+1} = Ax_i + Bu_i + w_i, \qquad A = \begin{bmatrix} 1 & T_s \\ -kT_s & 1 - dT_s \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ T_s \end{bmatrix}$$

We assume $-0.01 \leq \Delta k = k_{\text{true}} - k \leq 0.01$ and $-0.1 \leq \Delta d = d_{\text{true}} - d \leq 0.1$.

For the simulations we set $k = 0.3$, $d = 0.2$ and $k_{true} = 0.29$, $d_{true} = 0.3$.

1. We first investigate the robust MPC solution of Exercise 1 for this modified system. Note that robust guarantees designed for additive disturbances hold as long as

   $$x(k+1) - Ax(k) - Bu(k) \in \mathcal{W}$$

   where the controller designed was robust for $\mathcal{W} = \{Bw \,|\, |w| \leq 0.2\}$. Now we consider two sources of uncertainty: One is due to the model mismatch described previously, and the other are additional unmodeled disturbances $\bar{w}$

   (a) Compute the maximum $\bar{w}$ for which $w(k) \in \bar{\mathcal{W}} := \{Bw \,|\, |w| \leq \bar{w}\}$ ensures $x(k+1) - Ax(k) - Bu(k) + w(k) \in \mathcal{W}$ for all $x(k) \in \mathcal{X}$ and $u(k) \in \mathcal{U}$, i.e. how large the unstructured disturbance can be on top of the model mismatch such that the original robust guarantees apply.

   *Hint: Note that both disturbance and model error only affect the second state, i.e. the velocity.*

   (b) Load yesterday's robust MPC solution `RobustMPC.mat` and simulate the system with the robust MPC and adjusted noise level by drawing $w(k)$ uniformly from $\bar{\mathcal{W}}$. What do you notice?

2. Next, we set up a set membership estimator to infer the parameters $k$ and $d$ from state measurements. We set up the estimation problem as follows:

$$x(k+1) = Ax(k) + Bu(k) + \phi(x,u)\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} + w(k),\ \phi(x,u) = \left[[A_1,\ B_1]\begin{bmatrix}x\\ u\end{bmatrix},\ [A_2,\ B_2]\begin{bmatrix}x\\ u\end{bmatrix}\right]$$

where $\theta = [\theta_1,\ \theta_2]^T = [\Delta k,\ \Delta d]^T$ and $w(k) \in \bar{\mathcal{W}}$.

   (a) Set up the matrices $[A_i,\ B_i]$, $i = 1, 2$ as well as the initial parameter set $\theta \in \Omega_0 = \{\theta \mid -0.01 \leq \theta_1 \leq 0.01,\ -0.1 \leq \theta_2 \leq 0.1\}$.

   (b) Open the file `SetMembership.m` and complete the function `update(obj,xp,x,u)` by defining the set $\Delta_k$ (`D` in code) and performing the intersection $\Omega_{k+1} = \Omega_k \cap \Delta_k$.
   *Hint: $\phi(x,u)$ and $[A,\ B]$ are available as `obj.get_phixu(x,u)` and `obj.AB0` of the `SetMembership` class. You can access $H_w$ and $h_w$ as `obj.W.A` and `obj.W.b`*

   (c) Run the set-membership estimation on the previously generated trajectory.

3. In addition to the parameter set estimates, the provided `SetMembership` class returns a point estimate as the center of the bounding box of the current parameter set. In this part, we want to use this point estimate to design a *performance learning MPC*.

   (a) Open the file `PLMPC.m` and complete the optimization problem definition in the constructor of the performance learning MPC class. This includes the nominal state $z_{i+1} = Az_i + Bv_i$ w.r.t. which constraints are formulates as in the RobustMPC of Exercise 1, as well as a performance state $x_{i+1} = A_p x_i + B_p v_i$ w.r.t. which the objective is formulated. The matrices $A_p$ and $B_p$ are passed as parameters to the optimization problem.

   (b) Complete the *solve* method in the PLMPC.m by setting the current performance matrices $A_p$ and $B_p$ as the current point estimates.

   (c) Evalutate the performance learning MPC on the system.

4. Finally, we want to make use of the parameters set from the set-membership regression to reduce conservativeness w.r.t. the constraints. For this, the `SetMembership` class provides the function `estimateW_theta(X,U)` returning the set of possible model errors

$$\mathcal{W}_k^\theta = \left\{ w \;\middle|\; w = \phi(x,u)(\theta - \hat{\theta}_k),\ x \in \mathcal{X},\ u \in \mathcal{U},\ \theta \in \Omega \right\}$$

   *Note: There exist approaches which directly deal with the parametric uncertainty. Pointers:*

   - *Lorenzen et al. "Robust MPC with recursive model update", Automatica, 2019*
   - *Köhler et al, "Linear robust adaptive model predictive control: Computational complexity and conservatism," CDC, 2019*

   (a) Complete optimization problem the set-membership MPC class `SMMPC.m` to allow for parameterized state & input constraints

   (b) Complete the `solve` method by updating constraint tightening w.r.t. the current effective uncertainty $\mathcal{W}_k = \mathcal{W}_k^\theta \oplus \bar{\mathcal{W}}$

   (c) Evaluate the resulting set-membership-based MPC on the system.