RUHR-UNIVERSITÄT BOCHUM

Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries

FrontSeat Summer School on Optimization-Based Embedded Control Systems

Multi-Parametric Toolbox (MPT)

Day III – Morning Session

Juraj Holaza

STU in Bratislava

juraj.holaza@stuba.sk

September 3, 2025

MPT Introduction

$$
\begin{aligned}
\min_{u_0,\dots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2 \right) \\
\text{s.t.:} \quad & x_{k+1} = A x_k + B u_k \\
& x_k \in \mathcal{X} \\
& u_k \in \mathcal{U}
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
\min_{U} \quad & 0.5\, U^\intercal H U + \theta^\intercal F U + C_\mathrm{f} U \\
\text{s.t.:} \quad & G\, U \leq W + E\, \theta
\end{aligned}
$$

Can we reformulate this optimization problem within a minute?

MPT Introduction

The Multi-Parametric Toolbox (**MPT**):

- model predictive control,
- parametric optimization,
- computational geometry.

MPT Introduction

The Multi-Parametric Toolbox (**MPT**):

- model predictive control,
- parametric optimization,
- computational geometry.

MPT Introduction

$$
\begin{aligned}
\min_{u_0,\dots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) \\
\text{s.t.:} \quad & x_{k+1} = Ax_k + Bu_k \\
& x_k \in \mathcal{X} \\
& u_k \in \mathcal{U}
\end{aligned}
$$

MPT Introduction

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
           [0.5; 1]);
```

$$
\begin{aligned}
\min_{u_0,\dots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) \\
\text{s.t.:} \quad & \boxed{x_{k+1} = Ax_k + Bu_k} \\
& x_k \in \mathcal{X} \\
& u_k \in \mathcal{U}
\end{aligned}
$$

MPT Introduction

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right)$$

$$\text{s.t.:} \quad x_{k+1} = Ax_k + Bu_k$$

$$x_k \in \mathcal{X}$$

$$u_k \in \mathcal{U}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
            [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

MPT Introduction

$$\min_{u_0,\ldots,u_{N-1}} \boxed{\sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right)}$$

$$\begin{aligned} \text{s.t.:} \quad & x_{k+1} = Ax_k + Bu_k \\ & x_k \in \mathcal{X} \\ & u_k \in \mathcal{U} \end{aligned}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
              [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

MPT Introduction

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) \\
\text{s.t.:} \quad & x_{k+1} = Ax_k + Bu_k \\
& x_k \in \mathcal{X} \\
& u_k \in \mathcal{U}
\end{aligned}
$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
           [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

```
>> MPC = MPCController(model,N);
```

MPT Introduction

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right)$$

$$\text{s.t.:} \quad x_{k+1} = Ax_k + Bu_k$$

$$x_k \in \mathcal{X}$$

$$u_k \in \mathcal{U}$$

$$\boxed{x_N \in \mathcal{X}_\mathrm{N}}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
            [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

```
>> model.x.with('terminalSet');
>> model.x.terminalSet = model.LQRSet;
```

```
>> MPC = MPCController(model,N);
```

MPT Introduction

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) \boxed{+\|x_N\|_P^2}$$

$$\begin{aligned}
\text{s.t.:} \quad & x_{k+1} = Ax_k + Bu_k \\
& x_k \in \mathcal{X} \\
& u_k \in \mathcal{U} \\
& x_N \in \mathcal{X}_{\mathrm{N}}
\end{aligned}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
           [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

```
>> model.x.with('terminalSet');
>> model.x.terminalSet = model.LQRSet;
```

```
>> model.x.with('terminalPenalty');
>> P = model.LQRPenalty;
>> model.x.terminalPenalty = QuadFunction(P);
```

```
>> MPC = MPCController(model,N);
```

MPT Introduction

$$\min_{u_0,\dots,u_{N-1}} \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) + \|x_N\|_P^2$$

$$\begin{aligned} \text{s.t.:}\quad & x_{k+1} = Ax_k + Bu_k \\ & x_k \in \mathcal{X} \\ & u_k \in \mathcal{U} \\ & x_N \in \mathcal{X}_\mathrm{N} \\ & \boxed{u_k - u_{k-1} \geq \Delta u_{\min}} \end{aligned}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
           [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

```
>> model.x.with('terminalSet');
>> model.x.terminalSet = model.LQRSet;
```

```
>> model.x.with('terminalPenalty');
>> P = model.LQRPenalty;
>> model.x.terminalPenalty = QuadFunction(P);
```

```
>> model.u.with('DeltaMin');
>> model.u.deltaMin = -1;
```

```
>> MPC = MPCController(model,N);
```

MPT Introduction

The Multi-Parametric Toolbox (**MPT**):

- model predictive control,
- parametric optimization,
- computational geometry.

MPT Introduction

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \left(\|x_k\|_Q^2 + \|u_k\|_R^2\right) + \|x_N\|_P^2$$

$$\begin{aligned} \text{s.t.:} \quad & x_{k+1} = Ax_k + Bu_k \\ & x_k \in \mathcal{X} \\ & u_k \in \mathcal{U} \\ & x_N \in \mathcal{X}_\mathrm{N} \\ & u_k - u_{k-1} \geq \Delta u_{\min} \end{aligned}$$

```
>> model = LTISystem('A', [1, 1; 0, 1], 'B', ...
           [0.5; 1]);
```

```
>> model.u.min = [-1];
>> model.u.max = [ 1];
>> model.x.min = [-5; -5];
>> model.x.max = [ 5; 5];
```

```
>> model.x.penalty = QuadFunction(diag([1, 1]));
>> model.u.penalty = QuadFunction(diag([0.01]));
>> N = 5;
```

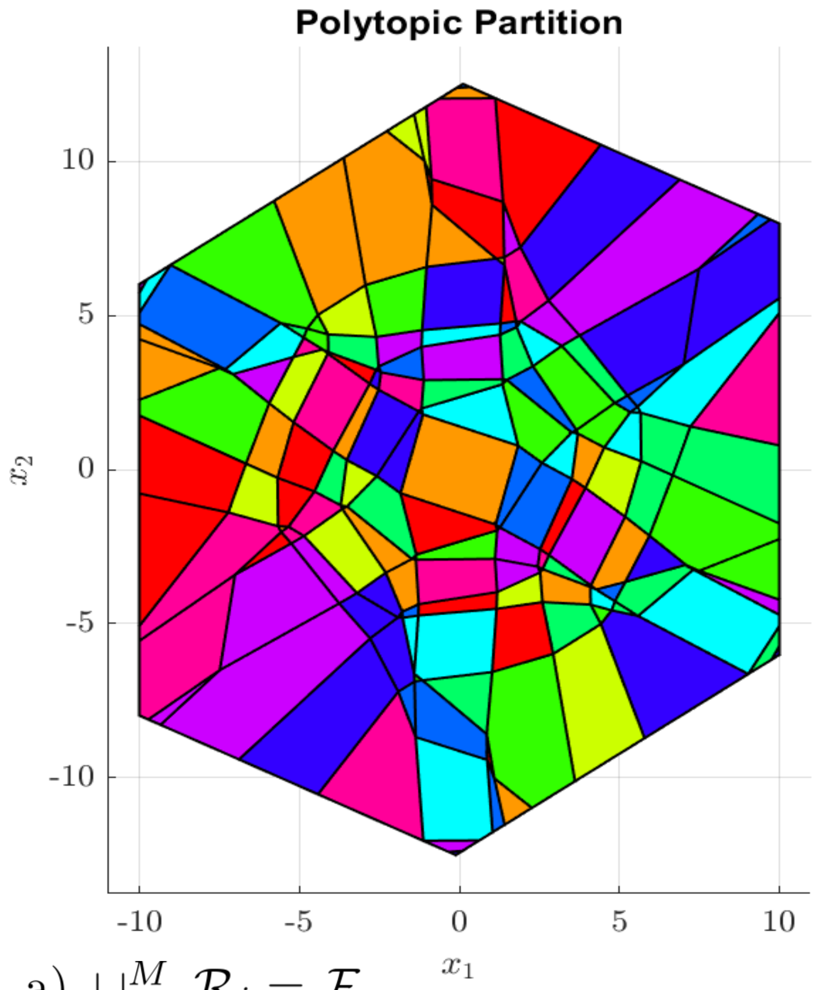
```
>> model.x.with('terminalSet');
>> model.x.terminalSet = model.LQRSet;
```

```
>> model.x.with('terminalPenalty');
>> P = model.LQRPenalty;
>> model.x.terminalPenalty = QuadFunction(P);
```

```
>> model.u.with('DeltaMin');
>> model.u.deltaMin = -1;
```

```
>> MPC = EMPCController(model,N);
```

MPT Introduction



a) $\cup_{i=1}^{M} \mathcal{R}_i = \mathcal{F}$,

b) $\mathcal{F} = \{x(t) \mid \exists u : u(k) \in \mathcal{U}, x(k) \in \mathcal{X}\}$,

c) $\mathrm{int}(\mathcal{R}_i) \cap \mathrm{int}(\mathcal{R}_j) = \varnothing, \forall i \neq j$,

MPT Workshop

MPT Workshop: Assignment and Templates

Assignment consists of:

- Module1_Basics
- Module2_CodeGen
- Module3_Stability
- Module4_Tracking
- Module5_AdvaceMPCSetup

MPT Workshop

https://github.com/holaza/FrontSeat_MPT.git

MPT Workshop: Assignment and Templates

A task from the assignment

Module 1: Basics

In this module, we will cover the fundamental formulation of Model Predictive Control (MPC) in Multi-parametric-Toolbox (MPT). For these tasks we will consider following MPC setup:

$$u^\star(x_0) = \arg\min_{u} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) \tag{1a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \; k = 0, \dots, N-1, \tag{1b}$$

$$y_k = Cx_k + Du_k, \; k = 0, \dots, N-1, \tag{1c}$$

$$u_{\min} \leq u_k \leq u_{\max}, \; k = 0, \dots, N-1, \tag{1d}$$

$$x_{\min} \leq x_k \leq x_{\max}, \; k = 0, \dots, N-1, \tag{1e}$$

with

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix},$$

$$u_{\min} = -1, \quad u_{\max} = 1, \quad x_{\min} = \begin{bmatrix} -5 \\ -5 \end{bmatrix}, \quad x_{\max} = \begin{bmatrix} 5 \\ 5 \end{bmatrix},$$

$$N = 3, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1.$$

1.) Formulate the MPC problem in the MPT framework using the provided template. Construct the MPC policy and evaluate it for a given initial condition $x_0 = [2\ 0]^\intercal$. Plot the open-loop sequence of states, inputs, and outputs. Create a closed-loop system composed of the online MPC controller and the prediction model. Simulate the evolution of the closed-loop system starting from $x_0 = [2\ 0]^\intercal$ for 20 steps. Plot the closed-loop profiles of states, outputs, and control inputs.

Q1: Are state and input constraints respected?

Q2: Why, in the open-loop response, the vector of states is longer than vector of outputs/inputs? What does the last state represent?

Associated MATLAB template

```matlab
%% Task 1a: Basic MPC formulation

% Defining model:
% x(k+1) = A*x(k) + B*u(k)
% y(k)   = C*x(k) + D*u(k)
% --------- Start Modifying Code Here -----------
% A = [...];
% B = [...];
% C = [...];
% D = [...];
% --------- End Modifying Code Here -----------
model = LTISystem('A', A, 'B', B, 'C', C, 'D', D, 'Ts', .1);

% Constraints
% --------- Start Modifying Code Here -----------
% model.x.max = [...];                  % constraints: x <= xmax
% model.x.min = [...];                  % constraints: x >= xmin
% model.u.max = [...];                  % constraints: u <= umax
% model.u.min = [...];                  % constraints: u >= umin
% --------- End Modifying Code Here -----------

% MPC setup
% --------- Start Modifying Code Here -----------
% Q = [...];                             % quadratic penalty x'*Q*x
% R = [...];                             % quadratic penalty u'*R*u
% N = ...;                               % prediction horizon
% --------- End Modifying Code Here -----------
model.x.penalty = QuadFunction(Q);       % quadratic penalty x'*Q*x
```

MPT Workshop: MPT Installation

MPT

https://www.mpt3.org/

MPT Workshop: MPT Installation

MPT3 Wiki

Multi-Parametric Toolbox 3

View Edit History Print

The Multi-Parametric Toolbox (or MPT for short) is an open source, Matlab-based toolbox for parametric optimization, computational geometry and model predictive control.

Installation

- Installation & updating instructions
- License
- How to cite MPT3

MPT Workshop: MPT Installation

MPT3 Wiki

View Edit History Print

Prerequisites

MATLAB R2011a or later.

Installation instructions

Plan A →

Automatic installation

Download and run the installation script install_mpt3.m in Matlab. The script basically executes the steps below and asks for installation directories.

Plan B →

Manual installation

1. Remove any previous installations of MPT2 and YALMIP from your path. If you do not have MPT2 or YALMIP installed, you can skip this step.

```
rmpath(genpath(fileparts(which('mpt_init'))))
rmpath(genpath(fileparts(which('yalmipdemo'))))
```


FrontSeat

MPT Workshop: MPT Installation (for MAC-M1)

Additional precompiled files for the M1 architecture:

MPT (M1)

https://github.com/MarekWadinger/tbx-index-arm64

MPT Workshop: Assignment and Templates

Available Modules:

- Module1_Basics
- Module2_CodeGen
- Module3_Stability
- Module4_Tracking
- Module5_AdvaceMPCSetup

MPT Workshop

https://github.com/holaza/FrontSeat_MPT.git

MPT

https://www.mpt3.org/

To extend functionality of MPT use command:

```
>> tbxmanager in mptplus
```