Model Predictive Control using MATLAB
1: Introduction and Preliminaries

by
Midhun T. Augustine

Overview

1. Introduction
   - MPC: Block Diagram
   - MPC: Terminologies
2. Preliminaries
   - General optimization problem
   - Convex optimization problem
   - Numerical optimization
3. MPC: Classifications

Introduction

Introduction

- MPC is a modern control approach which uses model based optimization for computing the control input.
- In MPC a model of the system is used to predict the future behaviour (states) of the system for a control input sequence.
- Also known as Receding Horizon Control (RHC).
- MPC: advantages over optimal control
  1. It gives closed-loop control schemes whereas optimal control mostly results in open-loop control schemes.
  2. MPC can handle complex systems such as nonlinear, higher-order, multi-variable, etc.
  3. MPC can incorporate constraints easily.

MPC: Block diagram

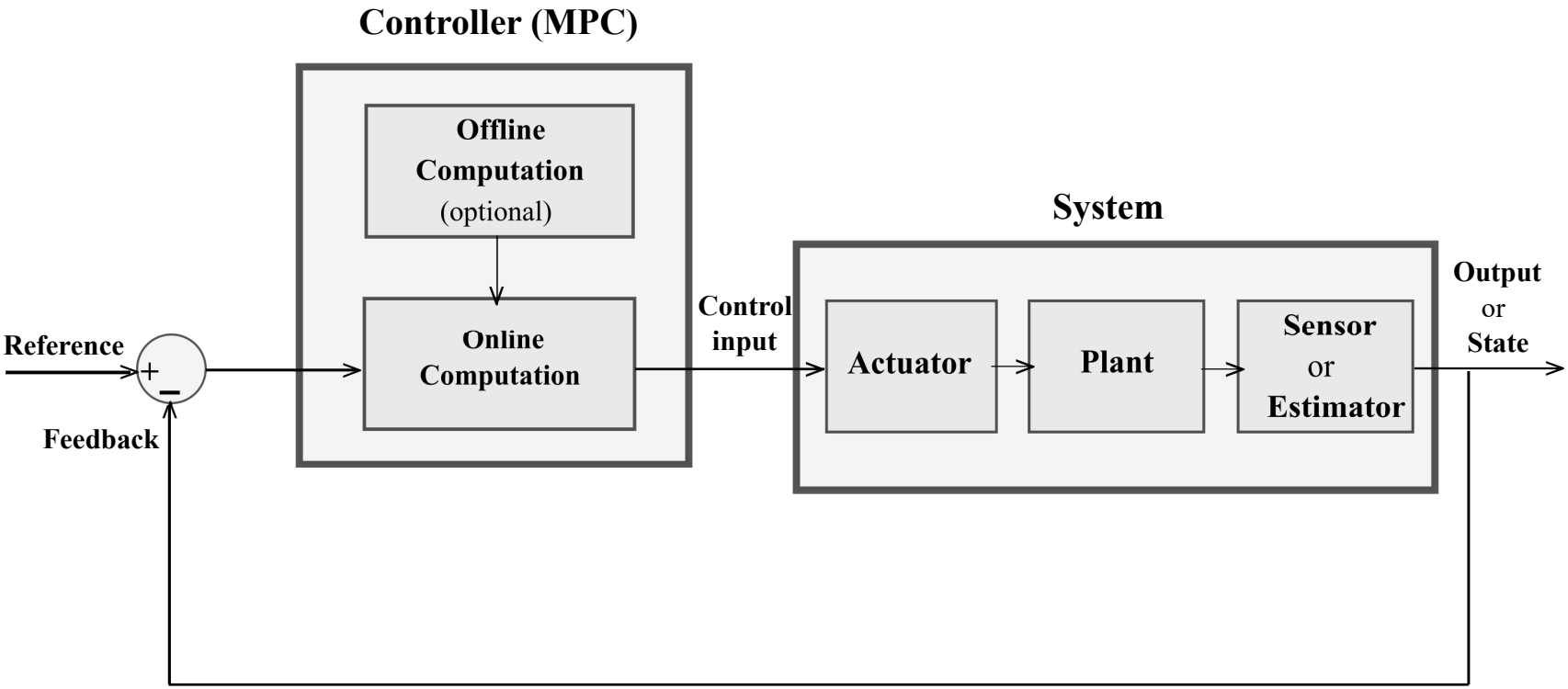


Figure 1: MPC: Block diagram

Notations

- $\mathbb{N}, \mathbb{Z}, \mathbb{R}$: Set of natural numbers, integers and real numbers.
- $\mathbb{R}^n$: $n-$ dimensional Euclidean space.
- $\mathbb{R}^{m \times n}$: Space of $m \times n$ real matrices.
- $\mathbf{A}, \mathbf{a}$: Matrix/vector $\mathbf{A}, \mathbf{a}$.
- A, a : Scalar A, a.
- $\mathbb{A}$: Set $\mathbb{A}$.
- $\mathbf{P} > 0$: Real symmetric positive definite matrix $\mathbf{P}$
- $\mathbf{P} \geq 0$: Real symmetric positive semidefinite matrix $\mathbf{P}$
- $\mathbf{I}, \mathbf{0}$: Identity matrix and zero matrix.

MPC: Terminologies

1. **Sampling time** (T): It is the time difference between two consecutive state measurements or control updates. In general $T \in \mathbb{R}^+$.
2. **Time horizon** (N_T): It is the number of time instants the control input is applied to the system. In general $N_T \in \mathbb{N}$.
3. **Prediction horizon** (N): It is the length of the prediction window over which the states are predicted and optimized. In general $N \in \mathbb{N}$ and usually $2 \leq N \leq N_T$.
4. **Control horizon** (N_C): It is the length of the control window in which the control input is optimized, and normally $N_C \leq N$.

MPC: Basic strategy

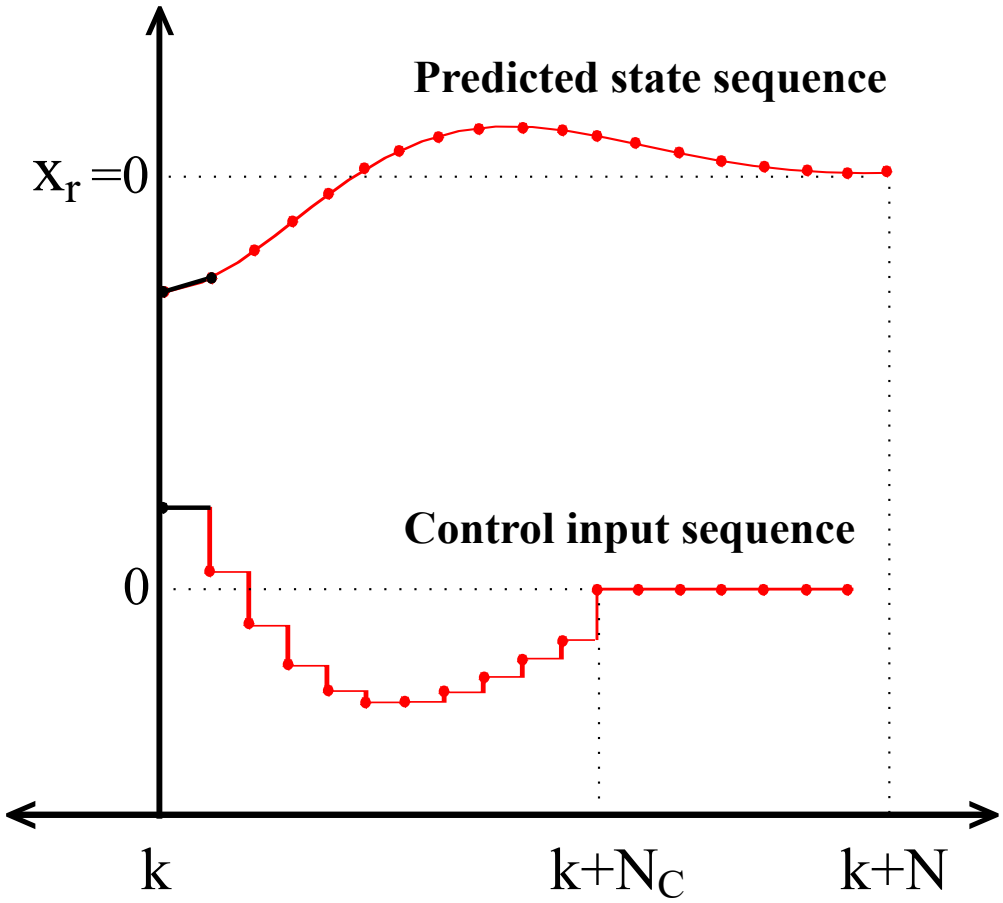


Figure 2: MPC: Basic strategy

Preliminaries

General optimization problem

- Optimization problem

$$\begin{aligned} \inf_{\mathbf{z}} \quad & f(\mathbf{z}) \quad subject\ to \\ & \mathbf{z} \in \mathbb{Z} \subseteq \mathbb{R}^p \end{aligned} \tag{1}$$

- $\mathbf{z}$: Decision vector $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_p \end{bmatrix}$
- f : Cost function. Some of the basic cost functions are
  1. $f(\mathbf{z}) = \mathbf{c}^T\mathbf{z}$: Linear cost function
  2. $f(\mathbf{z}) = \mathbf{z}^T\mathbf{H}\mathbf{z} + \mathbf{q}^T\mathbf{z} + r$: Quadratic cost function
- $\mathbb{Z}$: Constraint set.

General optimization problem

- Constrained set is usually defined using linear inequalities

$$\mathbb{Z} = \left\{ \mathbf{z} \in \mathbb{R}^2 : \begin{array}{l} z_1 \geq 0 \\ z_2 \geq 0 \\ z_1 + 2z_2 \leq 10 \\ 2z_1 + z_2 \leq 10 \end{array} \right\} \text{ which is equivalent to}$$

$$\mathbb{Z} = \{\mathbf{z} \in \mathbb{R}^2 : \mathbf{F}\mathbf{z} \leq \mathbf{g}\}, \qquad \mathbf{F} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

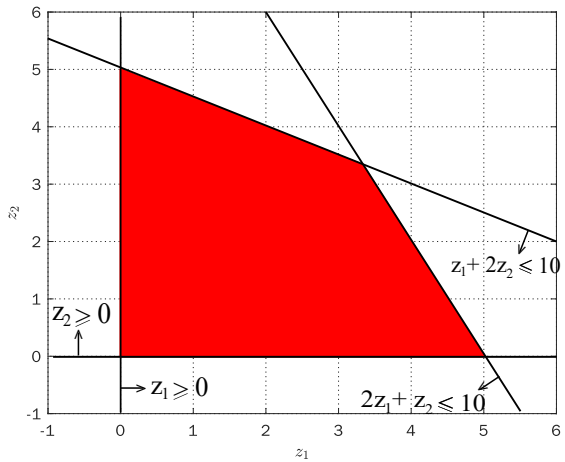


Figure 3: Constraint set (Polyhedron)

Convex optimization problem

- **Convex set**: A set $\mathbb{Z} \in \mathbb{R}^p$ is convex if

$$\lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2 \in \mathbb{Z} \tag{2}$$

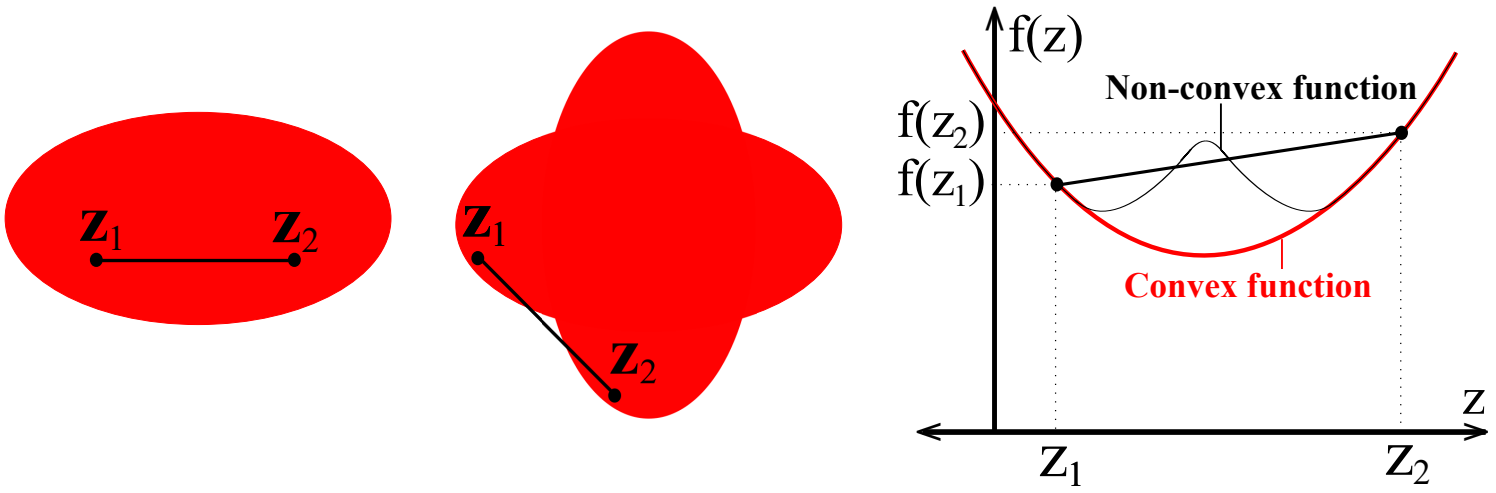
for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}$ and $\lambda \in [0, 1]$.

- **Convex function**: A function $f : \mathbb{Z} \to \mathbb{R}$ is convex if $\mathbb{Z}$ is convex and

$$f\big(\lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2\big) \leq \lambda f(\mathbf{z}_1) + (1-\lambda) f(\mathbf{z}_2) \tag{3}$$

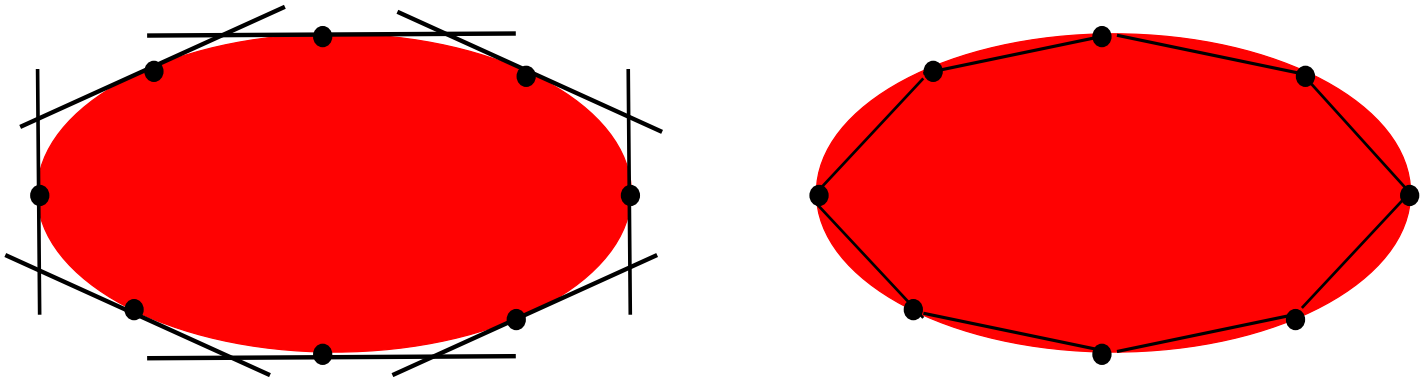
for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}$ and $\lambda \in [0, 1]$.

- **Convex optimization problem**: in which the cost function f is a convex function and the constraint set $\mathbb{Z}$ is a convex set.



(a) Convex set (b) Non-convex set (c) Convex function

Convex optimization problem



(a) Over approximation (b) Under approximation

Figure 5: Approximating a convex set with a polytope

Convex optimization problem: Examples

1. **Linear programming problem**: is of the form

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{c}^T\mathbf{z} \quad subject\ to \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{F}_{eq}\mathbf{z} = \mathbf{g}_{eq} \end{aligned} \tag{4}$$

2. **Quadratic programming problem**: is of the form

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T\mathbf{H}\mathbf{z} + \mathbf{q}^T\mathbf{z} + r \quad subject\ to \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{F}_{eq}\mathbf{z} = \mathbf{g}_{eq} \end{aligned} \tag{5}$$

Numerical optimization

- The two major approaches for numerical optimization
  1. **Iterative approach**: In which the elements of the decision vector are optimized together. Here the optimal decision vector is computed iteratively by starting with an initial guess which is then improved in each iterations.
  2. **Recursive approach**: In which the elements of the decision vector are optimized recursively, i.e., one at a time. The popular optimization algorithm which uses the recursive approach is the dynamic programming.

MPC: Classifications

MPC: Classifications

- Based on the type of system model used in optimization
  1. **Linear MPC**: The system model and the constraints are linear. The cost function can be linear or quadratic which results in linear or quadratic programming problems which are convex optimization problems.
  2. **Nonlinear MPC**: The system model is nonlinear and constraints are either linear or nonlinear. The cost function is usually chosen as a linear or quadratic function which results in a nonlinear programming problem which is non-convex.
- Based on the implementation
  1. **Implicit MPC**: This also known as the traditional MPC in which the control input at each time instant is computed by solving an optimization problem online.
  2. **Explicit MPC**: In this the online computation is reduced by transferring the optimization problem offline.

Thank you

Model Predictive Control using MATLAB
2: Linear MPC (LMPC)

by
Midhun T. Augustine

Overview

1. LMPC: Introduction
   - Constrained LQR (CLQR) Probelm
   - LMPC Problem

2. LMPC: Algorithm

LMPC: Introduction

System model

- Discrete-time linear time-invariant (LTI) system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{1}$$

- $k \in \mathbb{T} = \{0, 1, ..., N_T - 1\}$: discrete time instant.
- $\mathbf{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$: state vector.
- $\mathbf{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$: control input vector.
- $\mathbf{A} \in \mathbb{R}^{n \times n}$: system matrix.
- $\mathbf{B} \in \mathbb{R}^{n \times m}$: input matrix.
- $\mathbb{X}$, $\mathbb{U}$: constraint sets for the states and control inputs defined by

$$\begin{aligned} \mathbb{X} &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{F}_\mathbf{x}\mathbf{x} \leq \mathbf{g}_\mathbf{x}\} \\ \mathbb{U} &= \{\mathbf{u} \in \mathbb{R}^m : \mathbf{F}_\mathbf{u}\mathbf{u} \leq \mathbf{g}_\mathbf{u}\}. \end{aligned} \tag{2}$$

Constrained LQR (CLQR) Probelm

- Cost function

$$J = \mathbf{x}_{N_T}^T \mathbf{Q}_{N_T} \mathbf{x}_{N_T} + \sum_{k=0}^{N_T-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \tag{3}$$

  where $\mathbf{Q}_{N_T} \geq 0, \mathbf{Q} > 0$, $\mathbf{R} > 0$ are the weighting matrices.

- State and control sequence

$$\begin{aligned} \mathbf{X} &= \left(\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N_T}\right) \\ \mathbf{U} &= \left(\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{N_T-1}\right) \end{aligned} \tag{4}$$

Problem 1 (CLQR)

For the LTI system with the initial state $\boldsymbol{x}_0$, compute the control sequence $\boldsymbol{U}$ by solving the optimization problem

$$\begin{aligned} &\inf_{\boldsymbol{U}} \; J \quad \text{subject to} \\ &\quad \boldsymbol{U} \in \mathbb{U}^{N_T}, \quad \boldsymbol{X} \in \mathbb{X}^{N_T+1} \\ &\quad \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k, \quad k \in \mathbb{T}. \end{aligned} \tag{5}$$

LMPC Problem

- Cost function

$$J_k = \mathbf{x}_{k+N|k}^T \mathbf{Q}_N \mathbf{x}_{k+N|k} + \sum_{i=k}^{k+N-1} \mathbf{x}_{i|k}^T \mathbf{Q} \mathbf{x}_{i|k} + \mathbf{u}_{i|k}^T \mathbf{R} \mathbf{u}_{i|k} \tag{6}$$

- Predicted state and control sequence

$$\begin{aligned} \mathbf{X}_k &= \left(\mathbf{x}_{k|k}, \mathbf{x}_{k+1|k}, ..., \mathbf{u}_{k+N|k}\right) \\ \mathbf{U}_k &= \left(\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, ..., \mathbf{u}_{k+N-1|k}\right) \end{aligned} \tag{7}$$

Problem 2 (LMPC)

For the LTI system with the current state $\boldsymbol{x}_{k|k} = \boldsymbol{x}_k$, compute the control sequence $\boldsymbol{U}_k$, by solving the optimization problem

$$\begin{aligned} &\inf_{\boldsymbol{U}_k} \ J_k \quad \text{subject to} \\ &\quad \boldsymbol{U}_k \in \mathbb{U}^N, \ \ \boldsymbol{X}_k \in \mathbb{X}^{N+1}, \quad k \in \mathbb{T} \\ &\quad \boldsymbol{x}_{i+1|k} = \boldsymbol{A}\boldsymbol{x}_{i|k} + \boldsymbol{B}\boldsymbol{u}_{i|k}, \quad k \in \mathbb{T}, i = k, ..., k+N-1. \end{aligned} \tag{8}$$

LMPC: Algorithm

LMPC: Algorithm

- The solution of the state equation gives

$$\begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{x}_{k+1|k} \\ \vdots \\ \mathbf{x}_{k+N|k} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{k|k} \\ \mathbf{u}_{k+1|k} \\ \vdots \\ \mathbf{u}_{k+N-1|k} \end{bmatrix} \tag{9}$$

- By defining the following matrices

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{x}_{k+1|k} \\ \vdots \\ \mathbf{x}_{k+N|k} \end{bmatrix}, \mathbf{U}_k = \begin{bmatrix} \mathbf{u}_{k|k} \\ \mathbf{u}_{k+1|k} \\ \vdots \\ \mathbf{u}_{k+N-1|k} \end{bmatrix}, \ \mathbf{A_X} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \mathbf{B_U} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix} \tag{10}$$

  the equation (9) is rewritten as

$$\mathbf{X}_k = \mathbf{A_X}\mathbf{x}_k + \mathbf{B_U}\mathbf{U}_k \tag{11}$$

LMPC: Algorithm

- Similarly, by defining

$$\mathbf{Q_X} = \begin{bmatrix} \mathbf{Q} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{Q}_N \end{bmatrix}, \quad \mathbf{R_U} = \begin{bmatrix} \mathbf{R} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \dots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{R} \end{bmatrix} \tag{12}$$

the cost function can be represented in terms of $\mathbf{X}_k$ and $\mathbf{U}_k$ as

$$J_k = \mathbf{X}_k^T \mathbf{Q_X} \mathbf{X}_k + \mathbf{U}_k^T \mathbf{R_U} \mathbf{U}_k \tag{13}$$

- And by defining

$$\mathbf{F_X} = \begin{bmatrix} \mathbf{F_x} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F_x} & \dots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F_x} \end{bmatrix}, \quad \mathbf{g_X} = \begin{bmatrix} \mathbf{g_x} \\ \mathbf{g_x} \\ \vdots \\ \mathbf{g_x} \end{bmatrix}, \quad \mathbf{F_U} = \begin{bmatrix} \mathbf{F_u} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F_u} & \dots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F_u} \end{bmatrix}, \quad \mathbf{g_U} = \begin{bmatrix} \mathbf{g_u} \\ \mathbf{g_u} \\ \vdots \\ \mathbf{g_u} \end{bmatrix} \tag{14}$$

the constraints are represented in terms of $\mathbf{X}_k$ and $\mathbf{U}_k$ as

$$\begin{aligned} \mathbf{F_X} \mathbf{X}_k &\leq \mathbf{g_X} \\ \mathbf{F_U} \mathbf{U}_k &\leq \mathbf{g_U} \end{aligned} \tag{15}$$

LMPC: Algorithm

- Define

$$\mathbf{z} = \begin{bmatrix} \mathbf{X_k} \\ \mathbf{U_k} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{Q_X} & \mathbf{0} \\ \mathbf{0} & \mathbf{R_U} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F_X} & \mathbf{0} \\ \mathbf{0} & \mathbf{F_U} \end{bmatrix}$$
$$\mathbf{g} = \begin{bmatrix} \mathbf{g_X} \\ \mathbf{g_U} \end{bmatrix}, \quad \mathbf{F}_{eq} = \begin{bmatrix} \mathbf{I} & -\mathbf{B_U} \end{bmatrix}, \quad \mathbf{g}_{eq} = \mathbf{A_X x}_k \tag{16}$$

- Using this the MPC optimization problem is represented as

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T \mathbf{H} \mathbf{z} \quad \textit{subject to} \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{F}_{eq}\mathbf{z} = \mathbf{g}_{eq} \end{aligned} \tag{17}$$

  which is a quadratic programming problem.

- The control input with MPC is

$$\mathbf{u}_k = [\mathbf{U}_k^*]_1 = \mathbf{u}_{k|k}^*. \tag{18}$$

LMPC: Algorithm

Algorithm 1 : LMPC

Require $\mathbf{A}, \mathbf{B}, N_T, N, n, m, \mathbf{Q}, \mathbf{R}, \mathbf{Q}_{N_T}, \mathbf{F_x}, \mathbf{g_x}, \mathbf{F_u}, \mathbf{g_u}$
Initialize $\mathbf{x}_0, \mathbf{z}_0$
Construct $\mathbf{A_X}, \mathbf{B_U}, \mathbf{Q}_X, \mathbf{R_U}, \mathbf{H}, \mathbf{F}, \mathbf{g}$
for $k = 0$ *to* $N_T - 1$ **do**
    $\mathbf{x}_k = [\mathbf{X}]_{k+1}$ (obtain $\mathbf{x}_k$ from measurement/estimation)
    Compute $\mathbf{F}_{eq}, \mathbf{g}_{eq}$
    Compute $\mathbf{z}^* = \begin{bmatrix} \mathbf{X}_k^* \\ \mathbf{U}_k^* \end{bmatrix}$ by solving the optimization problem
    Apply $\mathbf{u}_k = [\mathbf{U}_k^*]_1$ to the system
    Update $\mathbf{z}_0 = \mathbf{z}^*$
end for

Thank you

Model Predictive Control using MATLAB
3: LMPC - Simulation results

by
Midhun T. Augustine

Overview

1. LMPC: Single input system

2. LMPC: Multi-input system

LMPC: Single input system

LMPC: Algorithm

Algorithm 1 : LMPC

Require $\mathbf{A}, \mathbf{B}, N_T, N, n, m, \mathbf{Q}, \mathbf{R}, \mathbf{Q}_{N_T}, \mathbf{F_x}, \mathbf{g_x}, \mathbf{F_u}, \mathbf{g_u}$
Initialize $\mathbf{x}_0, \mathbf{z}_0$
Construct $\mathbf{A_X}, \mathbf{B_U}, \mathbf{Q}_X, \mathbf{R_U}, \mathbf{H}, \mathbf{F}, \mathbf{g}$
for $k = 0$ *to* $N_T - 1$ **do**
    $\mathbf{x}_k = [\mathbf{X}]_{k+1}$ (obtain $\mathbf{x}_k$ from measurement/estimation)
    Compute $\mathbf{F}_{eq}, \mathbf{g}_{eq}$
    Compute $\mathbf{z}^* = \begin{bmatrix} \mathbf{X}_k^* \\ \mathbf{U}_k^* \end{bmatrix}$ by solving the optimization problem.
    Apply $\mathbf{u}_k = [\mathbf{U}_k^*]_1$ to the system.
    Update $\mathbf{z}_0 = \mathbf{z}^*$
end for

- MATLAB function for solving the constrained optimization problem

$$\mathbf{z}^* = \text{fmincon}(f, \mathbf{z}_0, \mathbf{F}, \mathbf{g}, \mathbf{F}_{eq}, \mathbf{g}_{eq}, \mathbf{lb}, \mathbf{ub}) \tag{1}$$

LMPC: Single input system

- System parameters

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0 \\ -1 & 1.5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix} \tag{2}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_2, \mathbf{R} = 1, \mathbf{x}_0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \tag{3}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4}$$

LMPC: Single input system

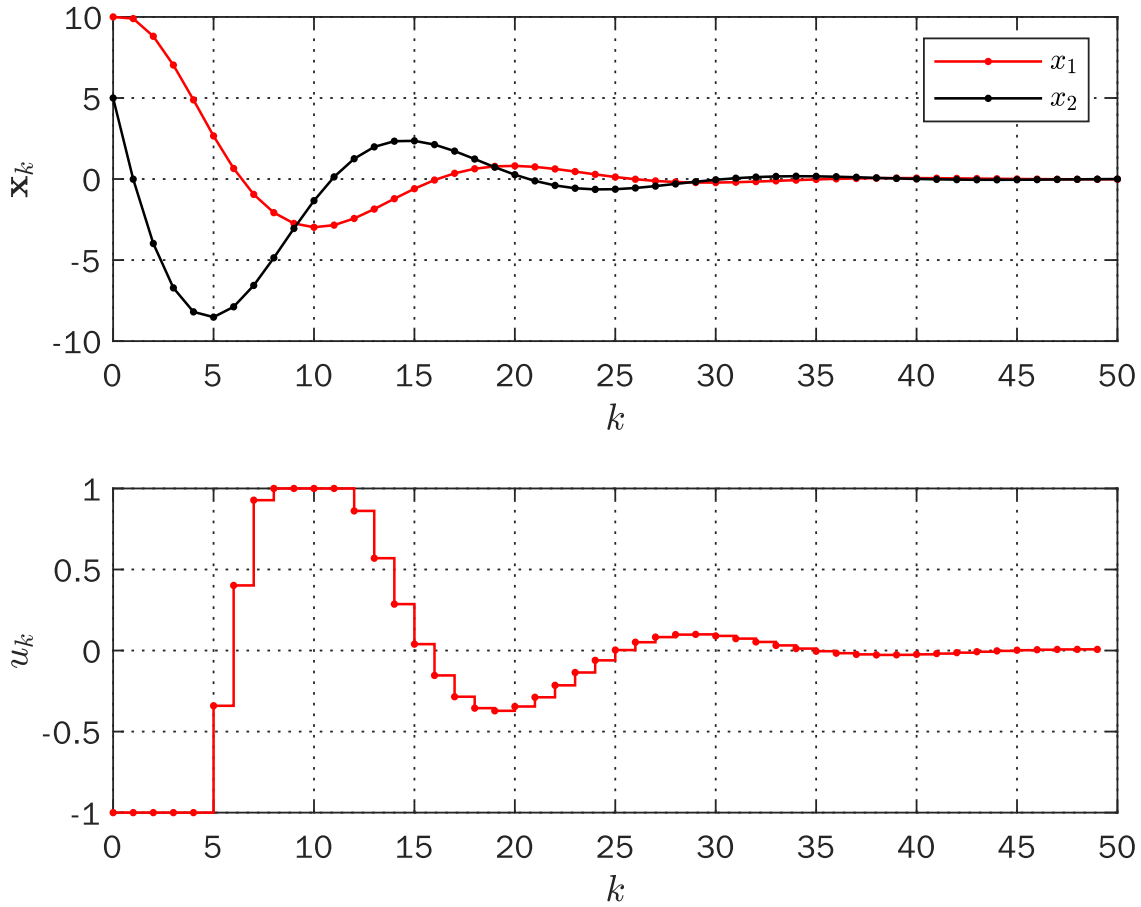


Figure 1: LMPC: Single input system

LMPC: Multi input system

LMPC: Multi input system

- System parameters

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0.2 & 0.1 \\ -0.4 & .8 & 0.1 \\ 0.3 & 0.2 & 0.5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{5}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_3, \mathbf{R} = \mathbf{I}_2, \mathbf{x}_0 = \begin{bmatrix} 10 \\ 5 \\ 2 \end{bmatrix} \tag{6}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{7}$$

LMPC: Multi input system

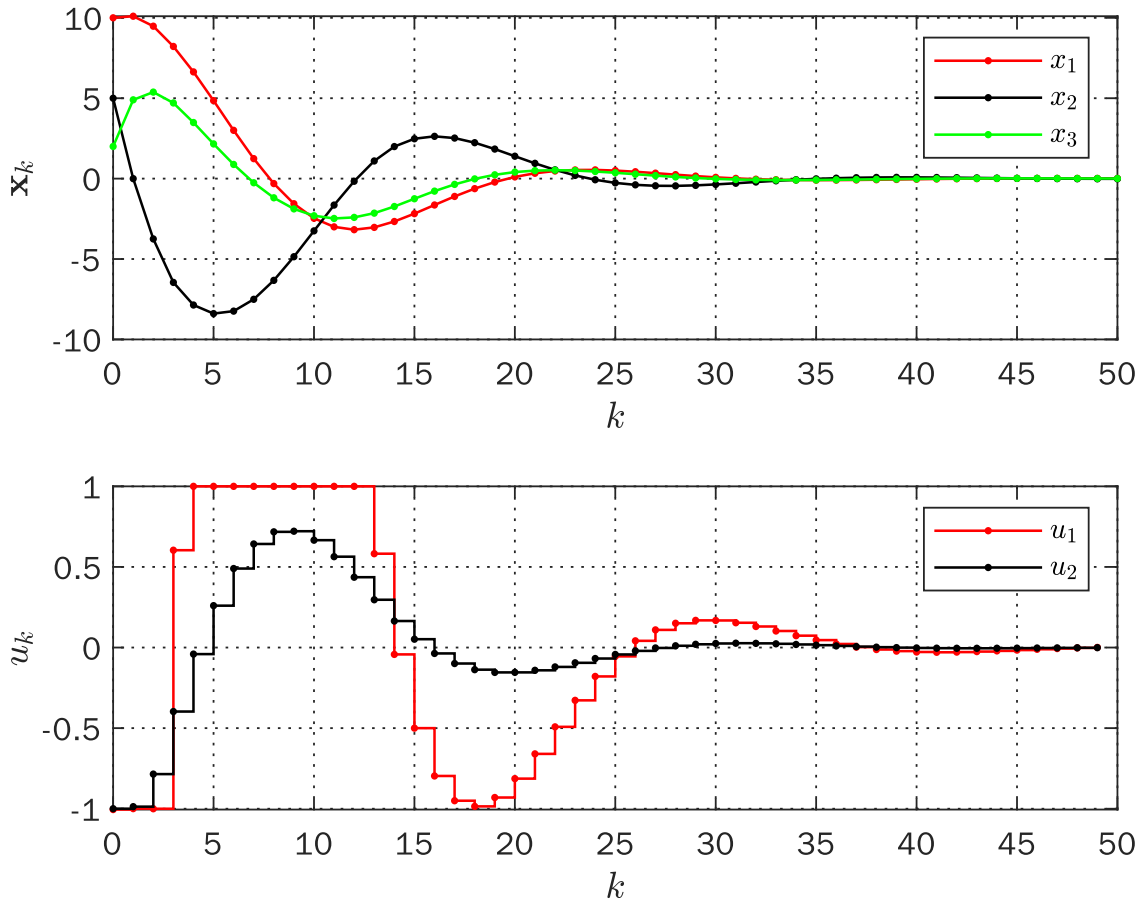


Figure 2: LMPC: Multi input system

Thank you

Model Predictive Control using MATLAB
4: LMPC - Reducing online computation

by
Midhun T. Augustine

Overview

1. LMPC: Reducing online computation

2. LMPC: Simulation results

LMPC: Reducing online computation

LMPC: Reducing online computation - method 1

- **Basic idea**: Reduce the number of variables in $\mathbf{z}$ by removing $\mathbf{X}_k$.
- $\mathbf{x}_k, \mathbf{x}_{i|k} \in \mathbb{R}^n$ contains n elements.
- $\mathbf{u}_k, \mathbf{u}_{i|k} \in \mathbb{R}^m$ contains m elements.
- $\mathbf{X}_k \in \mathbb{R}^{n(N+1)}$ contains $n(N+1)$ elements.
- $\mathbf{U}_k \in \mathbb{R}^{mN}$ contains mN elements.
- $\mathbf{z} = \begin{bmatrix} \mathbf{X}_k \\ \mathbf{U}_k \end{bmatrix} \in \mathbb{R}^{n(N+1)+mN}$ contains $n(N+1)+mN$ elements.
- Number of decision variables in $\mathbf{z}$: $n_{\mathbf{z}} = nN + mN$.

LMPC: Reducing online computation - method 1

- The state equation constraint

$$\mathbf{X}_k = \mathbf{A_X x}_k + \mathbf{B_U U}_k \tag{1}$$

- Substituting this in the cost function gives

$$\begin{aligned} J_k =& \left[\mathbf{A_X x}_k + \mathbf{B_U U}_k\right]^T \mathbf{Q_X} \left[\mathbf{A_X x}_k + \mathbf{B_U U}_k\right] + \mathbf{U}_k^T \mathbf{R_U U}_k \\ =& \mathbf{U}_k^T \left[\mathbf{B}_\mathbf{U}^T \mathbf{Q_X B_U} + \mathbf{R_U}\right] \mathbf{U}_k + 2\mathbf{x}_k^T \left[\mathbf{A}_\mathbf{X}^T \mathbf{Q_X B_U}\right] \mathbf{U}_k + \\ & \mathbf{x}_k^T \left[\mathbf{A}_\mathbf{X}^T \mathbf{Q_X A_X}\right] \mathbf{x}_k \\ =& \mathbf{U}_k^T \mathbf{H U}_k + \mathbf{q}_k^T \mathbf{U}_k + r_k \end{aligned} \tag{2}$$

where

$$\begin{aligned} \mathbf{H} &= \mathbf{B}_\mathbf{U}^T \mathbf{Q_X B_U} + \mathbf{R_U} \\ \mathbf{q}_k^T &= 2\mathbf{x}_k^T \mathbf{A}_\mathbf{X}^T \mathbf{Q_X B_U} \\ r_k &= \mathbf{x}_k^T \mathbf{A}_\mathbf{X}^T \mathbf{Q_X A_X x}_k \end{aligned} \tag{3}$$

LMPC: Reducing online computation - method 1

- Similarly the constraints becomes

$$\begin{aligned} &\mathbf{F_X}\left[\mathbf{A_X}\mathbf{x}_k + \mathbf{B_U}\mathbf{U}_k\right] \leq \mathbf{g_X} \implies \mathbf{F_X}\mathbf{B_U}\mathbf{U}_k \leq \mathbf{g_X} - \mathbf{F_X}\mathbf{A_X}\mathbf{x}_k \\ &\mathbf{F_U}\mathbf{U}_k \leq \mathbf{g_U} \end{aligned} \tag{4}$$

- Define $\mathbf{z} = \mathbf{U}_k$, $\mathbf{F} = \begin{bmatrix} \mathbf{F_X}\mathbf{B_U} \\ \mathbf{F_U} \end{bmatrix}$, $\mathbf{g} = \begin{bmatrix} \mathbf{g_X} - \mathbf{F_X}\mathbf{A_X}\mathbf{x}_k \\ \mathbf{g_U} \end{bmatrix}$ which results in the optimization problem

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T\mathbf{H}\mathbf{z} + \mathbf{q}_k^T\mathbf{z} + r_k \quad \textit{subject to} \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \end{aligned} \tag{5}$$

- Here the parameters $\mathbf{q}_k, r_k$ and $\mathbf{g}$ are functions of $\mathbf{x}_k$.
- Number of decision variables $n_{\mathbf{z}} = mN$.

LMPC: Reducing online computation - method 2

- **Basic idea**: Reduce the number of optimization variables by using a control horizon N_C lesser than the prediction horizon N.
- Define the control sequence as $\mathbf{U}_k = \left(\mathbf{u}_{k|k}, ..., \mathbf{u}_{k+N_C-1|k}, \mathbf{0}, ..., \mathbf{0}\right)$.
- Number of decision variables $n_{\mathbf{z}} = mN_C$.
- N_C is usually chosen as 2.

LMPC: Simulation results

LMPC: Simulation results

- Consider an LTI system

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0 \\ -1 & 1.5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix} \tag{6}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_2, \mathbf{R} = 1, \mathbf{x}_0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \tag{7}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{8}$$

LMPC: simulation results

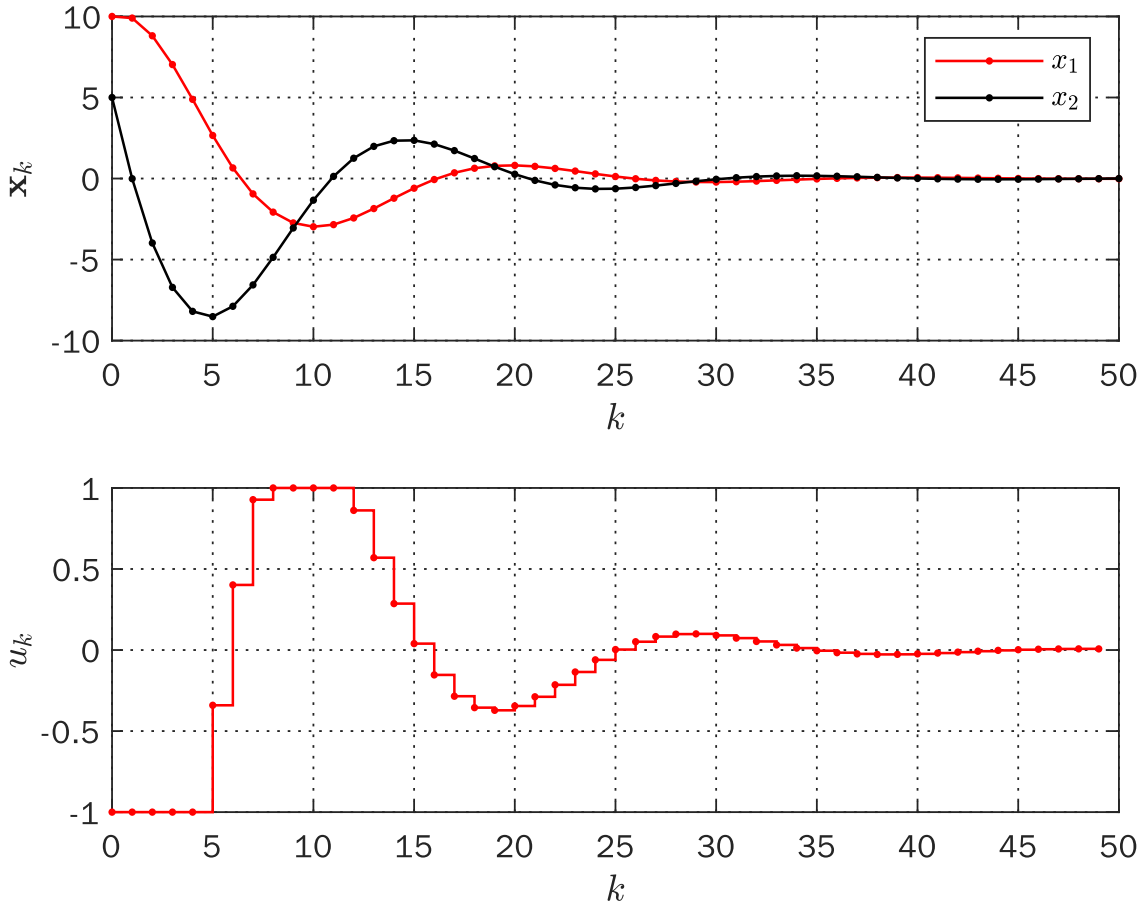


Figure 1: LMPC Response (Method 1)

LMPC: simulation results

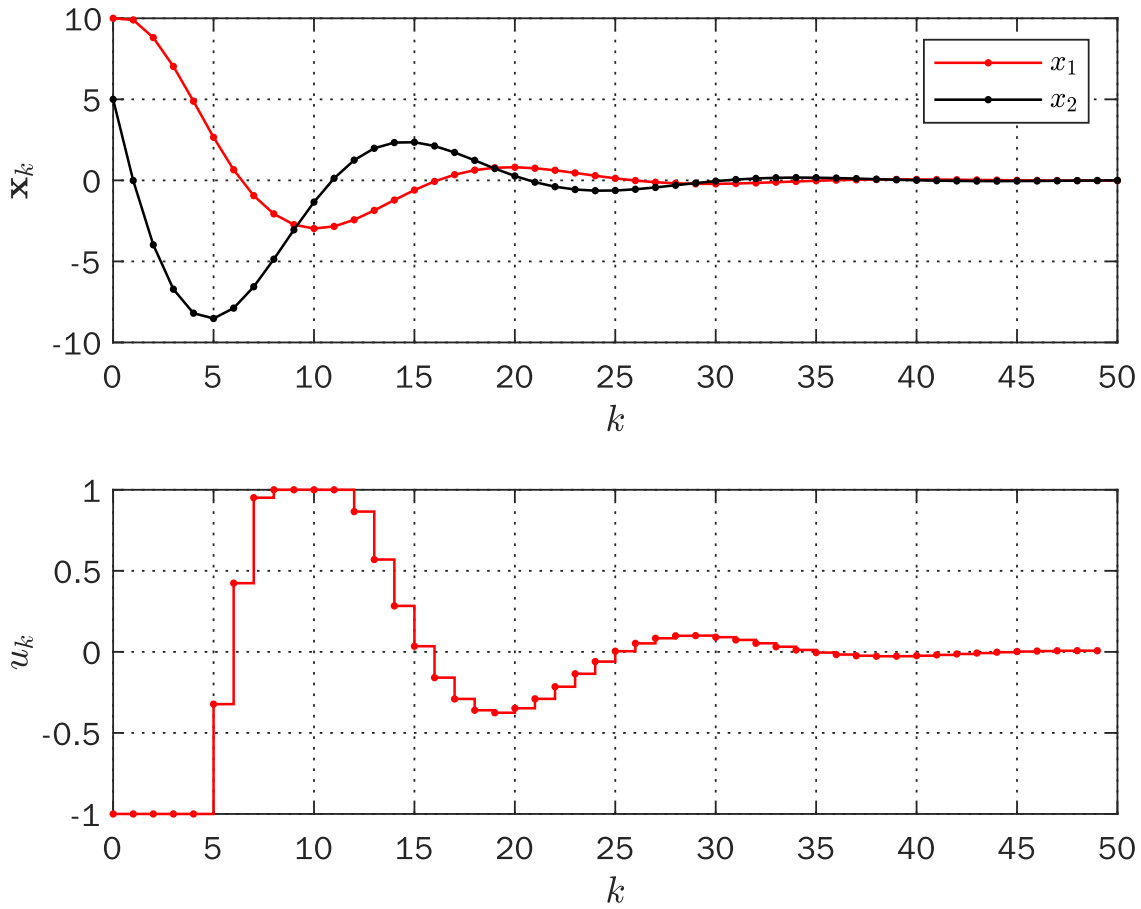


Figure 2: LMPC Response (Method 2 with $N_C = 2$)

Thank you

Model Predictive Control using MATLAB
5: LMPC - Set point tracking

by
Midhun T. Augustine

Overview

1. LMPC: Set-point tracking

2. LMPC: Simulation results

LMPC: Set-point tracking

Introduction

- Two major problems in MPC
  1. **Stabilization problem**: in which the reference $\mathbf{x}_r = 0$ and the objective is to drive the state to origin.
  2. **Set-point tracking problem**: in which the reference $\mathbf{x}_r \neq 0$, and the objective is to track the nonzero set point.
- For $\mathbf{x}_r \neq 0$, the steady state control input $\mathbf{u}_r \neq 0$. In steady state we have $\mathbf{x}_{k+1} = \mathbf{x}_k = \mathbf{x}_r$. Substituting this in the state equation gives

$$\begin{aligned} \mathbf{x}_r &= \mathbf{A}\mathbf{x}_r + \mathbf{B}\mathbf{u}_r \\ \implies \mathbf{u}_r &= \mathbf{B}^{-1}(\mathbf{I} - \mathbf{A})\mathbf{x}_r \end{aligned} \tag{1}$$

  where $\mathbf{B}^{-1}$ is the pseudo-inverse.

LMPC: Set-point tracking

- The set-point tracking can be transferred to a stabilization problem by using the error state and control vectors $\mathbf{x}_{e_k} = \mathbf{x}_k - \mathbf{x}_r, \mathbf{u}_{e_k} = \mathbf{u}_k - \mathbf{u}_r$.
- From state equation the error dynamics is obtained as

$$\begin{aligned} \mathbf{x}_{e_{k+1}} &= \mathbf{x}_{k+1} - \mathbf{x}_r = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k - \mathbf{x}_r \\ &= \mathbf{A}\mathbf{x}_k - \mathbf{A}\mathbf{x}_r + \mathbf{B}\mathbf{u}_k - \mathbf{x}_r + \mathbf{A}\mathbf{x}_r \\ &= \mathbf{A}[\mathbf{x}_k - \mathbf{x}_r] + \mathbf{B}[\mathbf{u}_k - \mathbf{B}^{-1}(\mathbf{I} - \mathbf{A})\mathbf{x}_r] = \mathbf{A}\mathbf{x}_{e_k} + \mathbf{B}\mathbf{u}_{e_k} \end{aligned} \tag{2}$$

- Using error state and control vectors the constraints can be rewritten as

$$\begin{aligned} \mathbf{F_x}\mathbf{x} \leq \mathbf{g_x} &\implies \mathbf{F_x}(\mathbf{x}_{e_k} + \mathbf{x}_r) \leq \mathbf{g_x} \implies \mathbf{F_x}\mathbf{x}_{e_k} \leq \mathbf{g_x} - \mathbf{F_x}\mathbf{x}_r \\ \mathbf{F_u}\mathbf{u} \leq \mathbf{g_u} &\implies \mathbf{F_u}(\mathbf{u}_{e_k} + \mathbf{u}_r) \leq \mathbf{g_u} \implies \mathbf{F_u}\mathbf{u}_{e_k} \leq \mathbf{g_u} - \mathbf{F_u}\mathbf{u}_r \end{aligned} \tag{3}$$

LMPC: set-point tracking

- Now the matrices $\mathbf{F_X}, \mathbf{g_X}, \mathbf{F_U}, \mathbf{g_U}$ can be defined as in stabilization case in which $\mathbf{g_x}, \mathbf{g_u}$ are replaced by $\mathbf{g}_x - \mathbf{F_x}\mathbf{x}_r, \mathbf{g}_u - \mathbf{F_u}\mathbf{u}_r$.
- Define $\mathbf{z} = \begin{bmatrix} \mathbf{X}_{e_k} \\ \mathbf{U}_{e_k} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k - \mathbf{X}_r \\ \mathbf{U}_k - \mathbf{U}_r \end{bmatrix}$ and the optimization problem becomes

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T \mathbf{H} \mathbf{z} \quad \textit{subject to} \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{F}_{eq}\mathbf{z} = \mathbf{g}_{eq} \end{aligned} \tag{4}$$

- The MPC control input for the set-point tracking problem is

$$\mathbf{u}_k = [\mathbf{U}^*_{e_k}]_1 + \mathbf{u}_r \tag{5}$$

LMPC: Simulation results

LMPC: Simulation results

- System parameters

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0 \\ -1 & 1.5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix} \tag{6}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_2, \mathbf{R} = 1, \mathbf{x}_0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \mathbf{x}_r = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, u_r = 0.59 \tag{7}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{8}$$

LMPC: set-point tracking

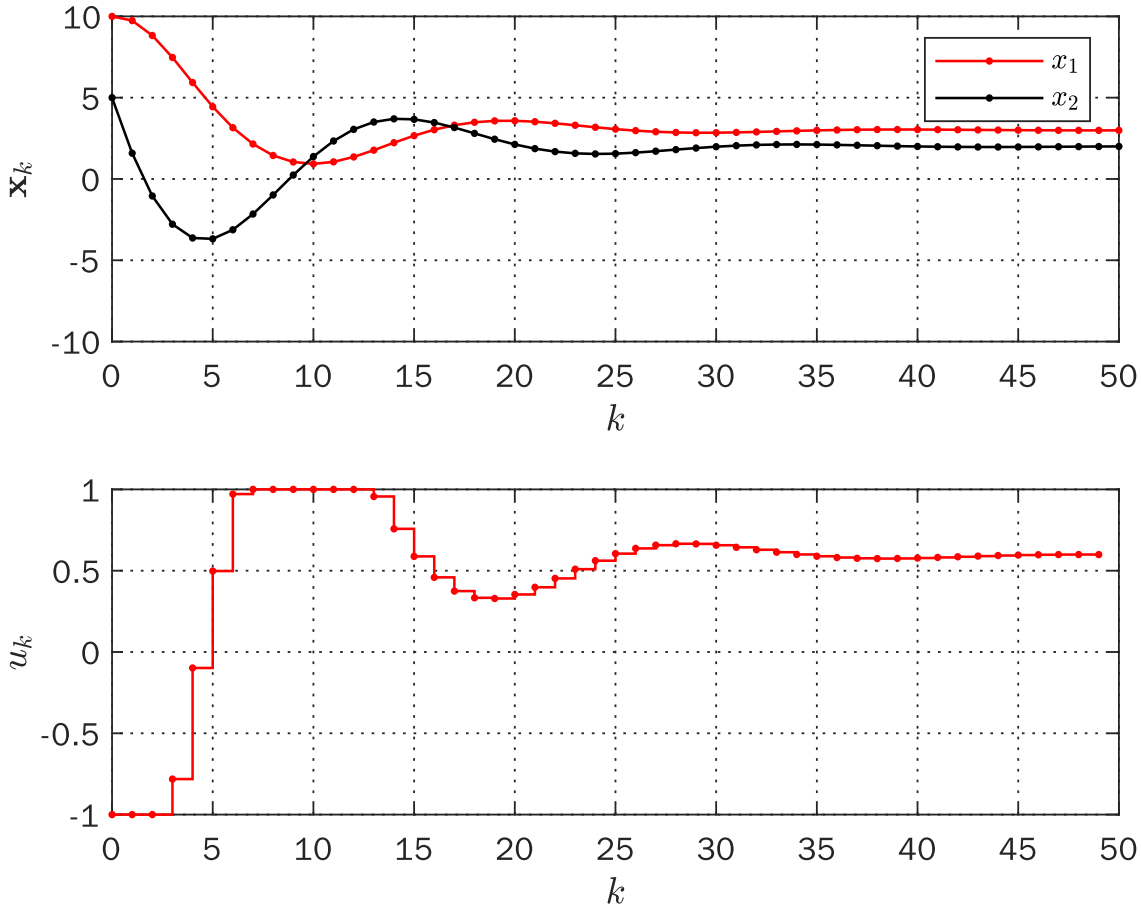


Figure 1: LMPC: set-point tracking

Thank you

Model Predictive Control using MATLAB
6: Nonlinear MPC (NMPC)

by
Midhun T. Augustine

Overview

1. NMPC: Introduction

2. NMPC: Algorithm

NMPC: Introduction

System model

- Discrete-time nonlinear system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \tag{1}$$

- $k \in \mathbb{T} = \{0, 1, ..., N_T - 1\}$: discrete time instant.
- $\mathbf{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$: state vector.
- $\mathbf{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$: control input vector.
- $\mathbf{f} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$: nonlinear mapping.
- $\mathbb{X}, \mathbb{U}$: constraint sets for the state and input vectors defined by

$$\begin{aligned} \mathbb{X} &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{F_x x} \leq \mathbf{g_x}\} \\ \mathbb{U} &= \{\mathbf{u} \in \mathbb{R}^m : \mathbf{F_u u} \leq \mathbf{g_u}\}. \end{aligned} \tag{2}$$

NMPC Problem

- Cost function

$$J_k = \mathbf{x}_{k+N|k}^T \mathbf{Q}_N \mathbf{x}_{k+N|k} + \sum_{i=k}^{k+N-1} \mathbf{x}_{i|k}^T \mathbf{Q} \mathbf{x}_{i|k} + \mathbf{u}_{i|k}^T \mathbf{R} \mathbf{u}_{i|k} \tag{3}$$

- Predicted state and control sequence

$$\begin{aligned} \mathbf{X}_k &= \left(\mathbf{x}_{k|k}, \mathbf{x}_{k+1|k}, ..., \mathbf{u}_{k+N|k}\right) \\ \mathbf{U}_k &= \left(\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, ..., \mathbf{u}_{k+N-1|k}\right) \end{aligned} \tag{4}$$

Problem 1 (NMPC)

For the nonlinear system with the current state $\boldsymbol{x}_{k|k} = \boldsymbol{x}_k$, compute the control sequence $\boldsymbol{U}_k$ by solving the optimization problem

$$\begin{aligned} &\inf_{\boldsymbol{U}_k} \ J_k \quad \text{subject to} \\ &\quad \boldsymbol{U}_k \in \mathbb{U}^N, \quad \boldsymbol{X}_k \in \mathbb{X}^{N+1}, \quad k \in \mathbb{T} \\ &\quad \boldsymbol{x}_{i+1|k} = \boldsymbol{f}(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}), \quad k \in \mathbb{T}, i = k, ..., k+N-1. \end{aligned} \tag{5}$$

NMPC: Algorithm

NMPC: Algorithm

- Using $\mathbf{X}_k$ and $\mathbf{U}_k$ the cost function and constraints for the NMPC can be rewritten as

$$J_k = \mathbf{X}_k^T \mathbf{Q}_\mathbf{X} \mathbf{X}_k + \mathbf{U}_k^T \mathbf{R}_\mathbf{U} \mathbf{U}_k \tag{6}$$

and

$$\begin{aligned} \mathbf{F}_\mathbf{X} \mathbf{X}_k &\leq \mathbf{g}_\mathbf{X} \\ \mathbf{F}_\mathbf{U} \mathbf{U}_k &\leq \mathbf{g}_\mathbf{U} \\ \mathbf{f}_{eq}(\mathbf{X}_k, \mathbf{U}_k) &= 0 \end{aligned} \tag{7}$$

where

$$\mathbf{f}_{eq}(\mathbf{X}_k, \mathbf{U}_k) = \begin{bmatrix} \mathbf{x}_{k|k} - \mathbf{x}_k \\ \mathbf{x}_{k+1|k} - \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_{k|k}) \\ \vdots \\ \mathbf{x}_{k+N|k} - \mathbf{f}(\mathbf{x}_{k+N-1|k}, \mathbf{u}_{k+N-1|k}) \end{bmatrix} \tag{8}$$

NMPC: Algorithm

- Define $\mathbf{z} = \begin{bmatrix} \mathbf{X_k} \\ \mathbf{U_k} \end{bmatrix}$ and $\mathbf{H}, \mathbf{F}, \mathbf{g}$ as in LMPC, the optimization problem is obtained as

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T \mathbf{H} \mathbf{z} \quad subject\ to \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{f}_{eq}(\mathbf{z}) = 0 \end{aligned} \tag{9}$$

- Here the equality constraint is nonlinear which makes the optimization problem a nonlinear programming problem.
- The control input with MPC is

$$\mathbf{u}_k = [\mathbf{U}_k^*]_1 = \mathbf{u}_{k|k}^*. \tag{10}$$

NMPC: Algorithm

Algorithm 1 : NMPC

Require $\mathbf{f}, N_T, N, n, m, \mathbf{Q}, \mathbf{R}, \mathbf{Q}_{N_T}, \mathbf{F_x}, \mathbf{g_x}, \mathbf{F_u}, \mathbf{g_u}$
Initialize $\mathbf{x}_0, \mathbf{z}_0$
Construct $\mathbf{Q}_X, \mathbf{R_U}, \mathbf{H}, \mathbf{F}, \mathbf{g}$
for $k = 0$ *to* $N_T - 1$ **do**
    $\mathbf{x}_k = [\mathbf{X}]_{k+1}$ (obtain $\mathbf{x}_k$ from measurement/estimation)
    Compute $\mathbf{z}^* = \begin{bmatrix} \mathbf{X}_k^* \\ \mathbf{U}_k^* \end{bmatrix}$ by solving the optimization problem
    Apply $\mathbf{u}_k = [\mathbf{U}_k^*]_1$ to the system
    Update $\mathbf{z}_0 = \mathbf{z}^*$
end for

Thank you

Model Predictive Control using MATLAB
7: NMPC - Simulation results

by
Midhun T. Augustine

Overview

1. NMPC: Algorithm

2. NMPC: Simple Pendulum

NMPC: Algorithm

NMPC: Algorithm

Algorithm 1 : NMPC

1: Require $\mathbf{f}, N_T, N, n, m, \mathbf{Q}, \mathbf{R}, \mathbf{Q}_{N_T}, \mathbf{F_x}, \mathbf{g_x}, \mathbf{F_u}, \mathbf{g_u}$
2: Initialize $\mathbf{x}_0, \mathbf{z}_0$
3: Construct $\mathbf{Q}_X, \mathbf{R_U}, \mathbf{H}, \mathbf{F}, \mathbf{g}$
4: **for** $k = 0$ *to* $N_T - 1$ **do**
5: $\quad \mathbf{x}_k = [\mathbf{X}]_{k+1}$ (obtain $\mathbf{x}_k$ from measurement/estimation)
6: $\quad$ Compute $\mathbf{z}^* = \begin{bmatrix} \mathbf{X}_k^* \\ \mathbf{U}_k^* \end{bmatrix}$ by solving the optimization problem
7: $\quad$ Apply $\mathbf{u}_k = [\mathbf{U}_k^*]_1$ to the system
8: $\quad$ Update $\mathbf{z}_0 = \mathbf{z}^*$
9: **end for**

- MATLAB function for solving the constrained optimization problem

$$\mathbf{z}^* = \text{fmincon}(f, \mathbf{z}_0, \mathbf{F}, \mathbf{g}, \mathbf{lb}, \mathbf{ub}, \mathbf{f}_{eq}) \tag{1}$$

NMPC: Simple Pendulum

Simple pendulum system

- State equation

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_{1_k} + T x_{2_k} \\ x_{2_k} + T\left(-\frac{g}{l} sin(x_{1_k}) - \frac{B}{Ml^2} x_{2_k} + \frac{1}{Ml^2} u_k \right) \end{bmatrix} \quad (2)$$

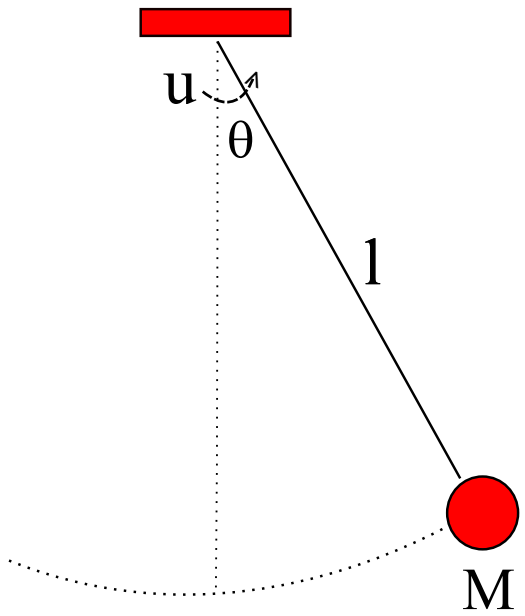


Figure 1: Simple pendulum

NMPC: Simulation results

- System parameters

$$M = 1, l = 1, B = 3, g = 9.8, T = 0.1 \tag{3}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_2, \mathbf{R} = 1, \mathbf{x}_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{4}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} \tag{5}$$

NMPC: Simple pendulum system

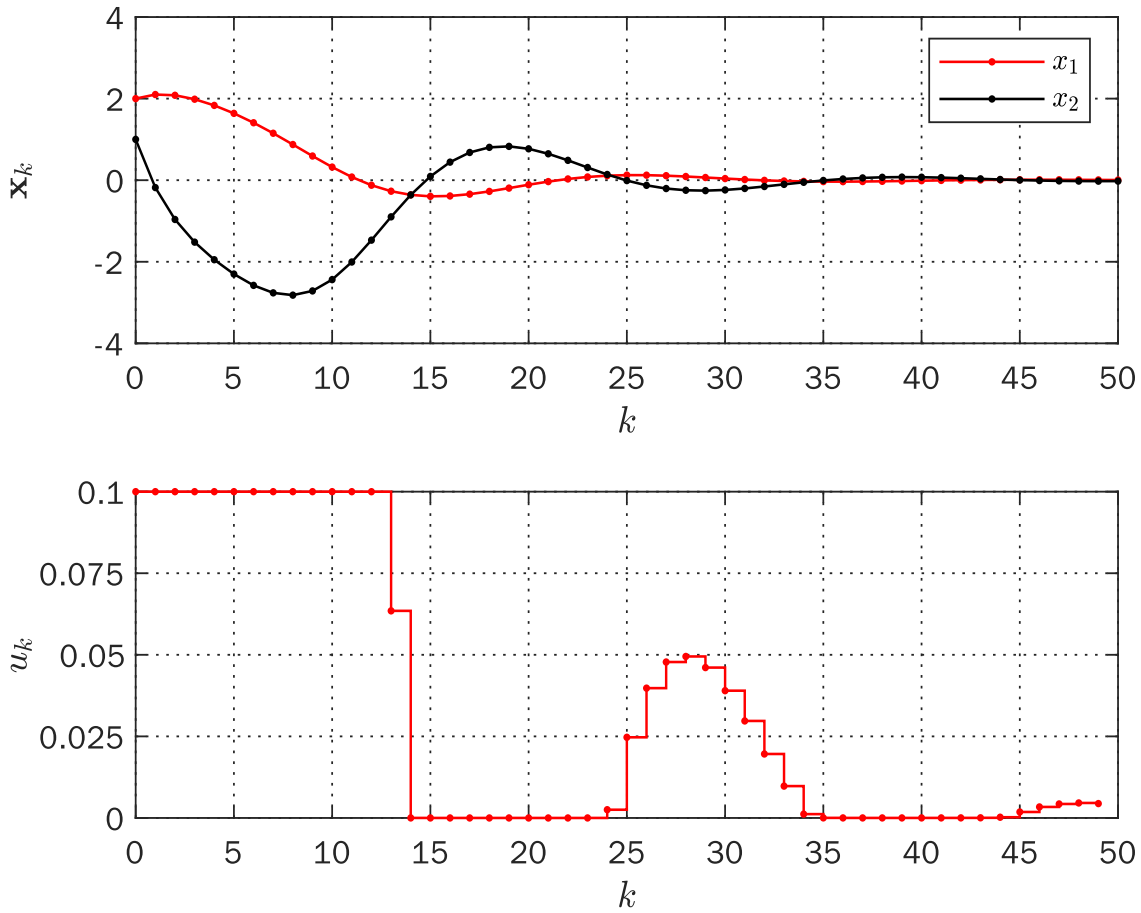


Figure 2: Simple pendulum: stabilization

Thank you

Model Predictive Control using MATLAB
8: NMPC - Set point tracking

by
Midhun T. Augustine

Overview

1. NMPC: Set-point tracking

2. NMPC: Simulation results

NMPC: Set-point tracking

NMPC: Set-point tracking

- **Set-point tracking problem**: in which the reference $\mathbf{x}_r \neq 0$.
- The reference value or steady state value of the control input $\mathbf{u}_r$ is computed by solving the steady-state equation

$$\mathbf{x}_r = \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r) \tag{1}$$

- Using the error state and control vectors $\mathbf{x}_{e_k} = \mathbf{x}_k - \mathbf{x}_r, \mathbf{u}_{e_k} = \mathbf{u}_k - \mathbf{u}_r$ the inequality constraints can be rewritten as in LMPC.
- Similarly the equality constraint becomes

$$\mathbf{x}_{e_{k+1}} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_r = \mathbf{f}(\mathbf{x}_{e_k} + \mathbf{x}_r, \mathbf{u}_{e_k} + \mathbf{u}_r) - \mathbf{x}_r \tag{2}$$

NMPC: Set-point tracking

- Define $\mathbf{z} = \begin{bmatrix} \mathbf{X}_{e_k} \\ \mathbf{U}_{e_k} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k - \mathbf{X}_r \\ \mathbf{U}_k - \mathbf{U}_r \end{bmatrix}$ and the optimization problem becomes

$$\begin{aligned} \inf_{\mathbf{z}} \quad & \mathbf{z}^T \mathbf{H} \mathbf{z} \quad \textit{subject to} \\ & \mathbf{F}\mathbf{z} \leq \mathbf{g} \\ & \mathbf{f}_{eq}(\mathbf{z}) = 0 \end{aligned} \tag{3}$$

- The MPC control input for the set-point tracking problem is obtained as

$$\mathbf{u}_k = [\mathbf{U}^*_{e_k}]_1 + \mathbf{u}_r \tag{4}$$

NMPC: Simulation results

NMPC: Simple pendulum system

- State equation

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_{1_k} + Tx_{2_k} \\ x_{2_k} + T\left(-\frac{g}{l} sin(x_{1_k}) - \frac{B}{Ml^2} x_{2_k} + \frac{1}{Ml^2} u_k \right) \end{bmatrix} \tag{5}$$

- The steady-state equation $\mathbf{x}_r = \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r)$ gives

$$\begin{aligned} x_{1_r} &= x_{1_r} + Tx_{2_r} \implies x_{2_r} = 0 \\ 0 &= T\left(-\frac{g}{l} sin(x_{1_r}) + \frac{1}{Ml^2} u_r \right) \implies u_r = Mgl\, sin(x_{1_r}) \end{aligned} \tag{6}$$

- System parameters

$$M = 1, l = 1, B = 3, g = 9.8, T = 0.1 \tag{7}$$

- Simulation parameters

$$N_T = 50, N = 5, \mathbf{Q} = \mathbf{I}_2, \mathbf{R} = 1, \mathbf{x}_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathbf{x}_r = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, u_r = 4.69 \tag{8}$$

- Constraint set parameters

$$\mathbf{F_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{g_x} = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} \quad \mathbf{F_u} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \mathbf{g_u} = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \tag{9}$$

Simple pendulum: set-point tracking

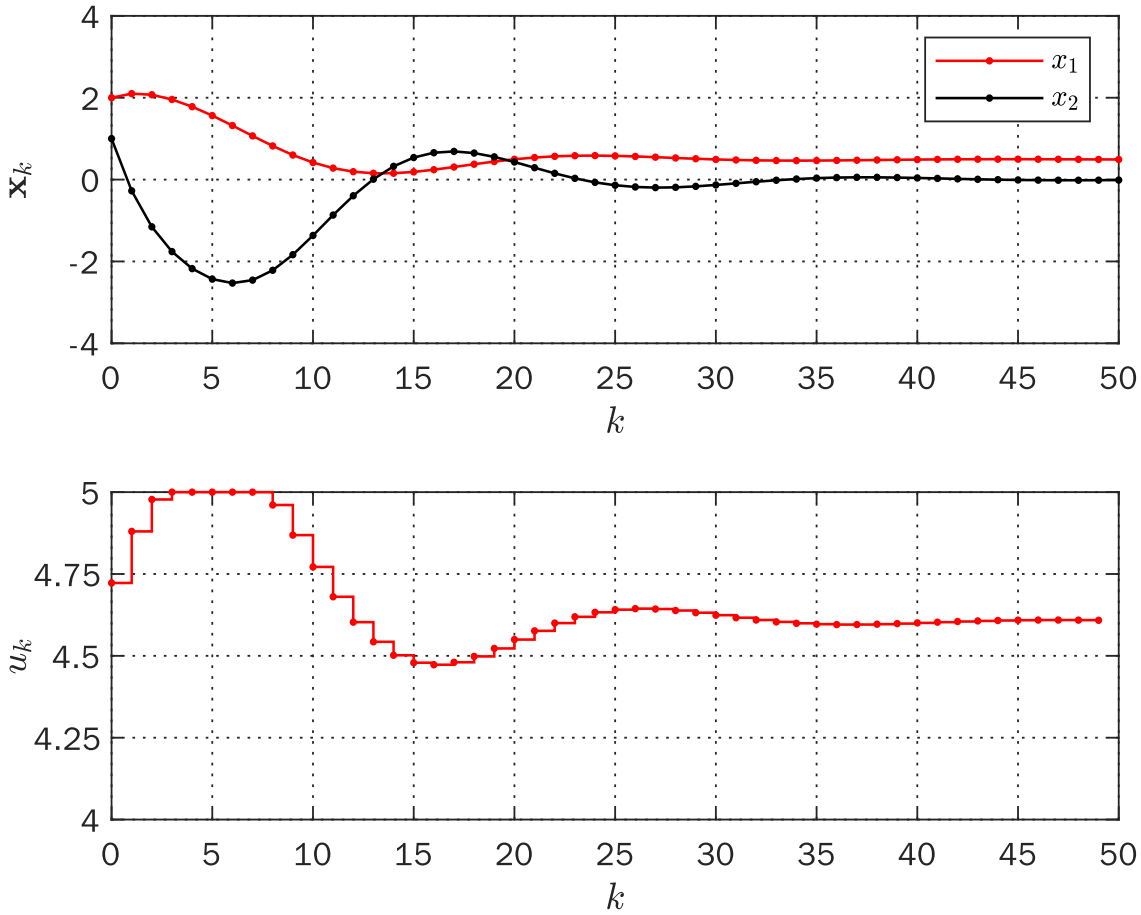


Figure 1: Simple pendulum: set-point tracking

Thank you

Model Predictive Control using MATLAB
9: Feasibility, Stability and Optimality

by
Midhun T. Augustine

Overview

1. Feasibility
   - Feasible set
2. Stability
   - Lyapunov Approach
3. Optimality

Feasibility

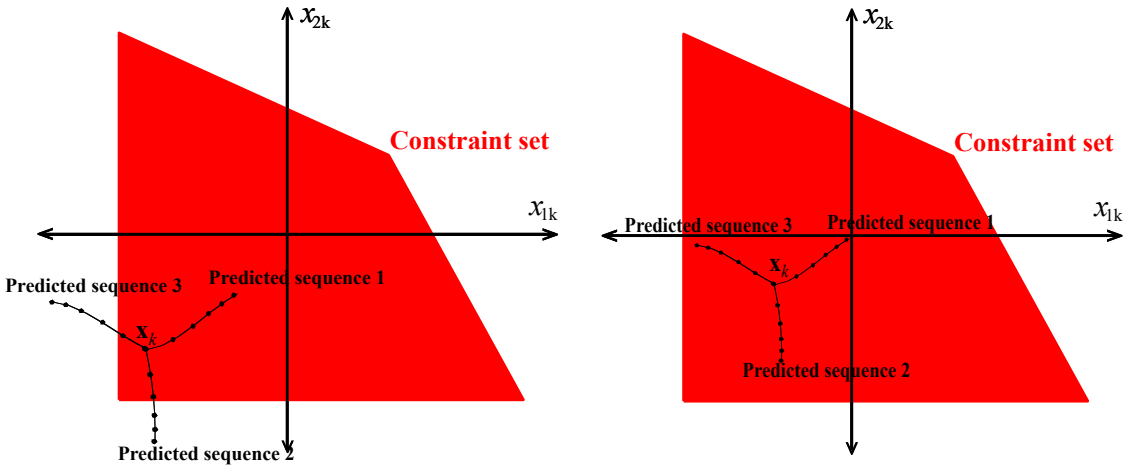
Feasible set

- **Feasible set of control sequences**: $\mathbb{U}_{fk} = \mathbb{U}_f(\mathbf{x}_k)$ is defined as

$$\mathbb{U}_{fk} = \{\mathbf{U}_k \in \mathbb{U}^N : \mathbf{X}_k(\mathbf{x}_k, \mathbf{U}_k) \in \mathbb{X}^{N+1}\} \tag{1}$$

- The number of elements in $\mathbb{U}_{fk} \subseteq \mathbb{U}^N$ decreases when $\mathbf{x}_k$ is closer to the boundary of $\mathbb{X}$.
- The MPC problem is said to be feasible for $\mathbf{x}_k \in \mathbb{X}$, if $\mathbb{U}_{fk}$ is nonempty.
- **Feasible set of states**: $\mathbb{X}_{fk} \subseteq \mathbb{X}$ is defined as

$$\mathbb{X}_{fk} = \{\mathbf{x}_k \in \mathbb{X} : \mathbb{U}_{fk} \neq \phi\} \tag{2}$$



Feasibility

- Let $\mathbb{X}_{fk}$ and $\mathbb{U}_{fk}$ are the feasible set of states and control sequences during time instant k.
- Then the MPC control law is computed by solving the optimization problem

$$\begin{aligned} \underset{\mathbf{U}_k \in \mathbb{U}_{f_k}}{inf} \quad & J_k(\mathbf{x}_k, \mathbf{U}_k) \quad subject\ to \\ & \mathbf{x}_{i+1|k} = \mathbf{f}(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad k \in \mathbb{T}, i = k, ..., k+N-1. \end{aligned} \tag{3}$$

- **Persistent feasibility**: feasibility of initial state $\mathbf{x}_0$ guarantees the feasibility of future states $\mathbf{x}_k, k = 1, 2, ..., N_T$,
  i.e. $\mathbb{U}_{f0} \neq \phi \implies \mathbb{U}_{fk} \neq \phi, \forall k = 1, 2, ..., N_T$.
- Persistent feasibility depends on the system dynamics, prediction horizon N, and the constrained sets $\mathbb{X}, \mathbb{U}$.

Stability

Stability

- **Lyapunov approach**: design the control scheme in such a way that the optimal cost function becomes a Lyapunov function, i.e. $V_k = J_k^*$ and it satisfies

$$\Delta V = J_{k+1}^*(\mathbf{x}_{k+1}) - J_k^*(\mathbf{x}_k) < 0 \tag{4}$$

- In general for stabilizable LTI systems, by properly selecting the terminal weighting matrix and constraints the value function for the MPC scheme can be made as a Lyapunov function.
- The terminal weighting matrix $\mathbf{Q}_N$ and terminal constraints $\mathbf{F}_{\mathbf{x}_N}, \mathbf{g}_{\mathbf{x}_N}$ can be easily incorporated in the MPC algorithm by adding them in $\mathbf{Q}_{\mathbf{X}}, \mathbf{F}_{\mathbf{X}}, \mathbf{g}_{\mathbf{X}}$.

Optimality

Optimality

- MPC usually results in suboptimal solution.
- As the prediction horizon increases the MPC control law becomes more optimal.
- In general as $N \to N_T$ the control law becomes optimal.
- N is selected based on a trade off between optimality and computation.

Thank you

Model Predictive Control using MATLAB
10: Further topics

by
Midhun T. Augustine

Overview

1. Advanced topics in MPC
2. Applications of MPC
3. References

Advanced topics in MPC

Advanced topics in MPC

1. **Adaptive MPC**: in which the model parameters are estimated online and the MPC is designed for the estimated model.
2. **Robust MPC**: considers MPC of uncertain systems with bounded uncertainties:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{1}$$

   where $\mathbf{w}_k \in \mathbb{W}$. $\mathbb{W}$ is a bounded set (usually polytope).

3. **Stochastic MPC**: considers MPC of uncertain systems with stochastic uncertainties:

$$\bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{u}_k, \bar{\mathbf{w}}_k) \tag{2}$$

   in which $\mathbf{w}_k, \mathbf{x}_0$ are random vectors.

Advanced topics in MPC

1. **Hybrid MPC**: deals with MPC of hybrid systems which are dynamical systems that has both continuous dynamics and discrete dynamics.
   Eg. Switched systems: which consists of a number of subsystems and a switching rule ($i = \sigma(k)$) that governs switching among the subsystems.

$$\mathbf{x}_{k+1} = \mathbf{f}_i(\mathbf{x}_k, \mathbf{u}_k) \tag{3}$$

   where $i \in \mathbb{M} = \{1, 2, ..., M\}$ and $\mathbb{M}$ is the set of subsystems.

2. **Distributed MPC**: considers MPC of distributed systems which consists of a number of local subsystems and a network over which the local systems communicate.

$$\mathbf{x}_{i_{k+1}} = \mathbf{f}(\mathbf{x}_{i_k}, \mathbf{u}_{i_k}) \tag{4}$$

   where $i \in \mathbb{M} = \{1, 2, ..., M\}$ and $\mathbb{M}$ is the set of subsystems

Applications of MPC

Applications of MPC

1. **Process industries**: initial applications of MPC mainly consists of the chemical and process industries. Initial versions of MPC: Generalized predictive control (GPC) and dynamic matrix control (DMC).
2. **Mechanical systems**: ability to handle constraints makes MPC suitable for controlling robots, automobiles and aerospace vehicles.
3. **Power converters**: recently MPC is extensively used in controlling power converters.
4. **Network systems**: one of the major future applications of MPC.

References

References

- F. Borrelli, A. Bemporad and M. Morari *"Predictive Control for Linear and Hybrid Systems"*, Cambridge University Press, 2017.
- L. Grune and J. Pannek *"Nonlinear Model Predictive Control: Theory and Algorithms"*, Springer, 2011.
- D. Luenberger and Y. Ye *"Linear and Nonlinear Programming: Fourth edition "*, Springer, 2008.
- D. Mayne, *"Model Predictive Control: Recent Developments and Future promise"*, Automatica, Issue 50, pp. 2967-2986, 2014.
- M. Guay, V. Adetola and D. Dehaan, *"Robust and Adaptive Model Predictive Control of Nonlinear Systems "*, The Institution of Engineering and Technology, 2015.
- B. Kouvaritakis and M. Cannon, *"Model Predictive Control: Classical, Robust and Stochastic"*, Springer, 2016.
- S. Rakovic and W. Levine, *"Handbook of Model Predictive Control"*, Springer, 2019.

Thank you

Contact: midhunta30@gmail.com