Netherlands Organisation for Scientific Research

Model-based Predictive Control (MPC)

Erjen Lefeber (TU/e), Marco Laumanns (IBM,ETHZ)

SCM-MPC Workshop, München

Technische Universiteit **Eindhoven** University of Technology

Where innovation starts

May 7, 2012

My background

(Nonlinear) Control Theory

What I like

Apply ideas/concepts from control theory in other fields

Four major problems

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Four major problems

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Four major problems

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Four major problems

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Technische Universiteit Eindhoven University of Technology

Example: Linear dynamics, discrete time

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \qquad\qquad x(0) = x_0 \\
y(k) &= Cx(k)
\end{aligned}
$$

Problem 1: Generate feasible reference trajectory

Determine $(u_r(k), x_r(k), y_r(k))$ satisfying

$$
\begin{aligned}
x_r(k+1) &= Ax_r(k) + Bu_r(k) \\
y_r(k) &= Cx_r(k)
\end{aligned}
$$

Typical approach: solve optimization problem

Example: Linear dynamics, discrete time

$$x(k+1) = Ax(k) + Bu(k) \qquad x(0) = x_0$$
$$y(k) = Cx(k)$$

Problem 1: Generate feasible reference trajectory

Determine $(u_r(k), x_r(k), y_r(k))$ satisfying

$$x_r(k+1) = Ax_r(k) + Bu_r(k)$$
$$y_r(k) = Cx_r(k)$$

Typical approach: solve optimization problem

Example: Linear dynamics, discrete time

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \qquad\qquad x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

Problem 1: Generate feasible reference trajectory

Determine $(u_r(k), x_r(k), y_r(k))$ satisfying

$$\begin{aligned} x_r(k+1) &= Ax_r(k) + Bu_r(k) \\ y_r(k) &= Cx_r(k) \end{aligned}$$

Typical approach: solve optimization problem

Design (static) state feedback controller

Given dynamics and reference

$$x(k+1) = Ax(k) + Bu(k) \qquad x_r(k+1) = Ax_r(k) + Bu_r(k)$$

Define tracking error and change of input:

$$\tilde{x}(k) = x(k) - x_r(k) \qquad \tilde{u}(k) = u(k) - u_r(k)$$

Resulting in error dynamics:

$$\begin{aligned}\tilde{x}(k+1) &= A\tilde{x}(k) + B\tilde{u}(k)\\ \tilde{y}(k) &= C\tilde{x}(k)\end{aligned}$$

Technische Universiteit Eindhoven University of Technology

Design (static) state feedback controller

Given dynamics and reference

$$x(k+1) = Ax(k) + Bu(k) \qquad x_r(k+1) = Ax_r(k) + Bu_r(k)$$

Define tracking error and change of input:

$$\tilde{x}(k) = x(k) - x_r(k) \qquad \tilde{u}(k) = u(k) - u_r(k)$$

Resulting in error dynamics:

$$\begin{aligned} \tilde{x}(k+1) &= A\tilde{x}(k) + B\tilde{u}(k) \\ \tilde{y}(k) &= C\tilde{x}(k) \end{aligned}$$

Design (static) state feedback controller

Given dynamics and reference

$$x(k+1) = Ax(k) + Bu(k) \qquad x_r(k+1) = Ax_r(k) + Bu_r(k)$$

Define tracking error and change of input:

$$\tilde{x}(k) = x(k) - x_r(k) \qquad \tilde{u}(k) = u(k) - u_r(k)$$

Resulting in error dynamics:

$$\begin{aligned} \tilde{x}(k+1) &= A\tilde{x}(k) + B\tilde{u}(k) \\ \tilde{y}(k) &= C\tilde{x}(k) \end{aligned}$$

Technische Universiteit Eindhoven University of Technology

Introduction to Control Theory: Problem 2

Design (static) state feedback controller

Problem: Determine $\tilde{u}([\tilde{x}(k)])$ such that $\lim_{k\to\infty} \tilde{x}(k) = 0$, where

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k) \qquad \tilde{x}(0) = \tilde{x}_0$$

Solution

Use $\tilde{u}(k) = -L\tilde{x}(k)$. Resulting closed loop dynamics:

$$\tilde{x}(k+1) = (A - BL)\tilde{x}(k) \qquad x(0) = \tilde{x}_0$$

Lemma: If $\operatorname{rank}\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n$ then eigenvalues of $A - BL$ can be placed arbitrarily.

Design (static) state feedback controller

Problem: Determine $\tilde{u}([\tilde{x}(k)])$ such that $\lim_{k\to\infty} \tilde{x}(k) = 0$, where

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k) \qquad \tilde{x}(0) = \tilde{x}_0$$

Solution

Use $\tilde{u}(k) = -L\tilde{x}(k)$. Resulting closed loop dynamics:

$$\tilde{x}(k+1) = (A - BL)\tilde{x}(k) \qquad x(0) = \tilde{x}_0$$

Lemma: If $\operatorname{rank}\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n$ then eigenvalues of $A - BL$ can be placed arbitrarily.

Design (static) state feedback controller

Problem: Determine $\tilde{u}([\tilde{x}(k)])$ such that $\lim_{k\to\infty}\tilde{x}(k)=0$, where

$$\tilde{x}(k+1)=A\tilde{x}(k)+B\tilde{u}(k) \qquad \tilde{x}(0)=\tilde{x}_0$$

Solution

Use $\tilde{u}(k)=-L\tilde{x}(k)$. Resulting closed loop dynamics:

$$\tilde{x}(k+1)=(A-BL)\tilde{x}(k) \qquad x(0)=\tilde{x}_0$$

Lemma: If $\operatorname{rank}\begin{bmatrix}B & AB & A^2B & \dots & A^{n-1}B\end{bmatrix}=n$ then eigenvalues of $A-BL$ can be placed arbitrarily.

Technische Universiteit Eindhoven University of Technology

Design (static) state feedback controller

Problem: Determine $\tilde{u}([\tilde{x}(k)])$ such that $\lim_{k\to\infty} \tilde{x}(k) = 0$, where

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k) \qquad \tilde{x}(0) = \tilde{x}_0$$

Solution

Use $\tilde{u}(k) = -L\tilde{x}(k)$. Resulting closed loop dynamics:

$$\tilde{x}(k+1) = (A - BL)\tilde{x}(k) \qquad x(0) = \tilde{x}_0$$

Lemma: If $\operatorname{rank}\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n$ then eigenvalues of $A - BL$ can be placed arbitrarily.

Technische Universiteit Eindhoven University of Technology

Design (static) state feedback controller

Given dynamics and reference

$$x(k+1) = Ax(k) + Bu(k) \qquad x_r(k+1) = Ax_r(k) + Bu_r(k)$$

we use the following controller:

$$u(k) = u_r(k) - L[x(k) - x_r(k)]$$

which guarantees that $\lim_{k\to\infty} x(k) - x_r(k) = 0$, provided L is properly chosen.

Design observer

Given dynamics

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \qquad\qquad x(0) = x_0 \\
y(k) &= Cx(k)
\end{aligned}
$$

Is it possible to reconstruct $x(k)$ from $u(k)$ and $y(k)$?

Solution

Let $\hat{x}(k)$ denote our estimate for $x(k)$:

$$
\begin{aligned}
\hat{x}(k+1) &= A\hat{x}(k) + Bu(k) \\
\hat{y}(k) &= C\hat{x}(k)
\end{aligned}
$$

Design observer

Given dynamics

$$x(k+1) = Ax(k) + Bu(k) \qquad\qquad x(0) = x_0$$
$$y(k) = Cx(k)$$

Is it possible to reconstruct $x(k)$ from $u(k)$ and $y(k)$?

Solution

Let $\hat{x}(k)$ denote our estimate for $x(k)$:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k)$$
$$\hat{y}(k) = C\hat{x}(k)$$

TU/e Technische Universiteit Eindhoven University of Technology

Design observer

Given dynamics

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \qquad\qquad x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

Is it possible to reconstruct $x(k)$ from $u(k)$ and $y(k)$?

Solution

Let $\hat{x}(k)$ denote our estimate for $x(k)$:

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

Technische Universiteit
Eindhoven
University of Technology

Design observer

Given dynamics

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \qquad\qquad x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

Is it possible to reconstruct $x(k)$ from $u(k)$ and $y(k)$?

Solution

Let $\hat{x}(k)$ denote our estimate for $x(k)$:

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

TU/e Technische Universiteit Eindhoven University of Technology

Design observer

Given dynamics and observer

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \qquad x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

For observer error $\bar{x}(k) = x(k) - \hat{x}(k)$, we obtain

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) - K\bar{y}(k) = [A - KC]\bar{x}(k) \\ \bar{y}(k) &= C\bar{x}(k) \end{aligned}$$

Technische Universiteit Eindhoven University of Technology

Design observer

Given dynamics and observer

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) & x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

For observer error $\bar{x}(k) = x(k) - \hat{x}(k)$, we obtain

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) - K\bar{y}(k) = [A - KC]\bar{x}(k) \\ \bar{y}(k) &= C\bar{x}(k) \end{aligned}$$

Technische Universiteit Eindhoven University of Technology

Design (dynamic) output feedback controller

Given dynamics and reference

$$x(k+1) = Ax(k) + Bu(k) \qquad x_r(k+1) = Ax_r(k) + Bu_r(k)$$

we use the following controller:

$$\begin{aligned} u(k) &= u_r(k) - L[\hat{x}(k) - x_r(k)] \\ \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

which guarantees that $\lim_{k\to\infty} x(k) - x_r(k) = 0$, provided K and L are properly chosen.

Four major problems

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Linear Quadratic Gaussian control (Problem 4)

System

$$X(k+1) = A(k)X(k) + B(k)U(k) + V(k)$$
$$Y(k) = C(k)X(k) + W(k)$$

with $V(k), W(k)$ Gaussian white noise (cov. matrices $\Sigma_v(k), \Sigma_w(k)$).

Objective

Minimize

$$J = E\left(X(N)^T Q(N) X(N) + \sum_{k=0}^{N-1} X(k)^T Q(k) X(k) + U(k)^T R(k) U(k)\right)$$

where $Q(k) \geq 0$, $R(k) > 0$.

Linear Quadratic Gaussian control (Problem 4)

System

$$X(k+1) = A(k)X(k) + B(k)U(k) + V(k)$$
$$Y(k) = C(k)X(k) + W(k)$$

with $V(k), W(k)$ Gaussian white noise (cov. matrices $\Sigma_v(k), \Sigma_w(k)$).

Objective

Minimize

$$J = E\left(X(N)^T Q(N) X(N) + \sum_{k=0}^{N-1} X(k)^T Q(k) X(k) + U(k)^T R(k) U(k) \right)$$

where $Q(k) \geq 0$, $R(k) > 0$.

Solution

Controller

$$\begin{aligned} U(k) &= -L(k)\hat{X}(k) \\ \hat{X}(k+1) &= A(k)\hat{X}(k) + B(k)U(k) + K(k)[Y(k) - C(k)\hat{X}(k)] \end{aligned}$$

Linear Quadratic Gaussian control (Problem 4)

Solution

Controller

$$
\begin{aligned}
U(k) &= -L(k)\hat{X}(k) \\
\hat{X}(k+1) &= A(k)\hat{X}(k) + B(k)U(k) + K(k)[Y(k) - C(k)\hat{X}(k)] \\
K(k) &= A(k)P(k)C(k)^T[C(k)P(k)C(k)^T + \Sigma_w(k)]^{-1} \\
P(k+1) &= A(k)\Gamma[C(k), P(k), \Sigma_w(k)]A(k)^T + \Sigma_v(k) \\
L(k) &= [B(k)^T S(k+1)B(k) + R(k)]^{-1}B(k)^T S(k+1)A(k) \\
S(k) &= A(k)^T\Gamma[B(k)^T, S(k+1), R(k)]A(k) + Q(k)
\end{aligned}
$$

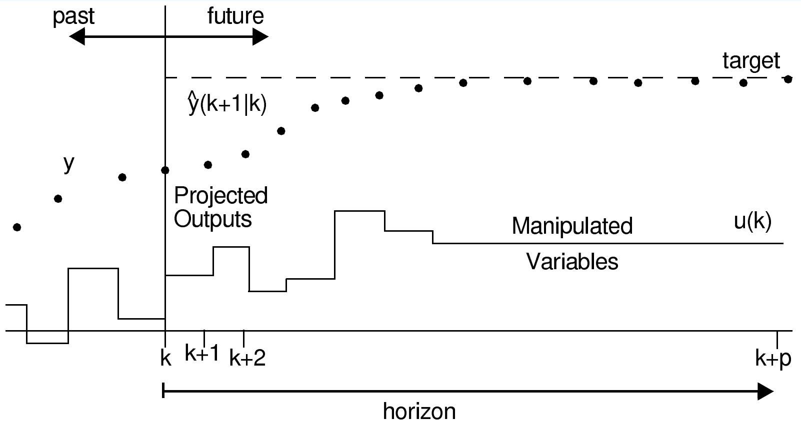
$S(N) = Q(N)$, $P(0) = E(X_0X_0^T)$, $\hat{x}(0) = E(X_0)$. Furthermore,

$$
\Gamma[F, G, H] = G - GF^T(FGF^T + H)^{-1}FG
$$

Technische Universiteit Eindhoven University of Technology

Model based predictive control

Receding horizon



First era: industrial success stories

1950s Various oil and petrochemical industries: optimal process settings computed every 15-20 minutes, implemented by manual operators.

60s and 70s Feedback controller from repeatedly using recomputed open loop controllers (Lee and Markus). Repeatedly solving Problem 1.

- Deterministic (without any disturbance model)
- Lack of stability guarantees
- Lack of systematic tuning guidelines

First era: industrial success stories

1950s Various oil and petrochemical industries: optimal process settings computed every 15-20 minutes, implemented by manual operators.

60s and 70s Feedback controller from repeatedly using recomputed open loop controllers (Lee and Markus). Repeatedly solving Problem 1.

- Deterministic (without any disturbance model)
- Lack of stability guarantees
- Lack of systematic tuning guidelines

Technische Universiteit Eindhoven University of Technology

First era: industrial success stories

1950s Various oil and petrochemical industries: optimal process settings computed every 15-20 minutes, implemented by manual operators.

60s and 70s Feedback controller from repeatedly using recomputed open loop controllers (Lee and Markus). Repeatedly solving Problem 1.

- Deterministic (without any disturbance model)
- Lack of stability guarantees
- Lack of systematic tuning guidelines

TU/e Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

Consider system $x(k+1) = Ax(k) + Bu(k)$.
At each time: measure (or estimate) state x_0 and solve

$$\min_{u(0),\ldots u(N-1)} \hat{x}(N)^T Q_N x(N) + \sum_{k=0}^{N-1} \left[x(k)^T Q x(k) + u(k)^T R u(k) \right]$$

$$\begin{aligned} \text{s.t.}\quad & x(k+1) = Ax(k) + Bu(k) \qquad x(0) = x_0 \\ & u(k) \in \mathbb{U} \\ & x(k) \in \mathbb{X} \qquad x_(N) \in \mathbb{X}_N \end{aligned}$$

where $\mathbb{U}$, $\mathbb{X}$, $\mathbb{X}(p)$ convex compact sets containing 0.

Result: feedback $u(x_0)$ (online calculation).

Second era: founding of MPC theory

Consider system $x(k+1) = Ax(k) + Bu(k)$.
At each time: measure (or estimate) state x_0 and solve

$$\min_{u(0),\dots u(N-1)} \hat{x}(N)^T Q_N x(N) + \sum_{k=0}^{N-1} \left[x(k)^T Q x(k) + u(k)^T R u(k) \right]$$

$$\text{s.t.} \quad x(k+1) = Ax(k) + Bu(k) \qquad x(0) = x_0$$

$$u(k) \in \mathbb{U}$$

$$x(k) \in \mathbb{X} \qquad x_(N) \in \mathbb{X}_N$$

where $\mathbb{U}$, $\mathbb{X}$, $\mathbb{X}(p)$ convex compact sets containing 0.

Result: feedback $u(x_0)$ (online calculation).

Example

Consider dynamics

$$x(k+1) = 4x(k) + u(k) \qquad x(k|k) = x_0$$

Horizon of 1: $\min_{u(k|k)} x(k+1|k)^2 + u(k|k)^2$

$$\min_{u(k|k)} [4x_0 + u(k|k)]^2 + u(k|k)^2 = 16x_0^2 + 8x_0 u(k|k) + 2u(k|k)^2$$

Optimal solution: $u(k|k) = \frac{-8x_0}{2 \cdot 2} = -2x_0$

Closed-loop system: $x(k+1) = 4x(k) - 2x(k) = 2x(k)$ Unstable!

Technische Universiteit Eindhoven University of Technology

Model based predictive control

Example

Consider dynamics

$$x(k+1) = 4x(k) + u(k) \qquad x(k|k) = x_0$$

Horizon of 1: $\min_{u(k|k)} x(k+1|k)^2 + u(k|k)^2$

$$\min_{u(k|k)} [4x_0 + u(k|k)]^2 + u(k|k)^2 = 16x_0^2 + 8x_0 u(k|k) + 2u(k|k)^2$$

Optimal solution: $u(k|k) = \frac{-8x_0}{2 \cdot 2} = -2x_0$

Closed-loop system: $x(k+1) = 4x(k) - 2x(k) = 2x(k)$ Unstable!

Model based predictive control

Example

Consider dynamics

$$x(k+1) = 4x(k) + u(k) \qquad x(k|k) = x_0$$

Horizon of 1: $\min\limits_{u(k|k)} x(k+1|k)^2 + u(k|k)^2$

$$\min_{u(k|k)} [4x_0 + u(k|k)]^2 + u(k|k)^2 = 16x_0^2 + 8x_0 u(k|k) + 2u(k|k)^2$$

Optimal solution: $u(k|k) = \frac{-8x_0}{2 \cdot 2} = -2x_0$

Closed-loop system: $x(k+1) = 4x(k) - 2x(k) = 2x(k)$ Unstable!

TU/e Technische Universiteit Eindhoven University of Technology

Example

Consider dynamics

$$x(k+1) = 4x(k) + u(k) \qquad x(k|k) = x_0$$

Horizon of 1: $\min\limits_{u(k|k)} x(k+1|k)^2 + u(k|k)^2$

$$\min_{u(k|k)} [4x_0 + u(k|k)]^2 + u(k|k)^2 = 16x_0^2 + 8x_0 u(k|k) + 2u(k|k)^2$$

Optimal solution: $u(k|k) = \frac{-8x_0}{2 \cdot 2} = -2x_0$

Closed-loop system: $x(k+1) = 4x(k) - 2x(k) = 2x(k)$ Unstable!

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T P x$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

Technische Universiteit Eindhoven University of Technology

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T P x$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T Px$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T P x$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T P x$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

TU/e Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

Observation 1: Infinite horizon results in stabilizing controller

Observation 2: After finite amount of time: solution remains unconstrained

Idea: Properly select terminal costs and horizon

Main steps

1. Solve infinite horizon LQR problem: $u = Kx$, $V = x^T Px$
2. Determine *maximal output admissible set*: $\mathbb{X}_N$
3. Determine N s.t. $x(N) \in \mathbb{X}_N$

TU/e Technische Universiteit Eindhoven University of Technology

Model based predictive control

Second era: founding of MPC theory

- Robust MPC (next slide)
- Nonlinear MPC

Third era: Diversification through fast MPC

- MPC for hybrid systems and systems with logical constraints
- Explicit MPC (mpLP,mpQP)
- Fast optimization
- Application (mechanical and electronic systems)

Technische Universiteit Eindhoven University of Technology

Second era: founding of MPC theory

- Robust MPC (next slide)
- Nonlinear MPC

Third era: Diversification through fast MPC

- MPC for hybrid systems and systems with logical constraints
- Explicit MPC (mpLP,mpQP)
- Fast optimization
- Application (mechanical and electronic systems)

Model based predictive control

Robust MPC

Dynamics

$$x(k+1) = Ax(k) + Bu(k) + Ed(k) \qquad d(k) \in \mathbb{D} = \{d : Ld \le l\}$$

$$J^{*(k)}(x^{(k)}) = \min_{u^{(k)}} J^{(k)}(x^{(k)}, u^{(k)})$$

$$\text{s.t.} \left\{ \begin{array}{l} Fx^{(k)} + Gu^{(k)} \le g \\ Ax^{(k)} + Bu^{(k)} \in \mathbb{X}^{(k)} \end{array} \right\} \forall d^{(k)} \in \mathbb{D}$$

$$J^{(k)}(x^{(k)}, u^{(k)}) = \max_{d^{(k)} \in \mathbb{D}} \|Qx^{(k)}\|_1 + \|Ru^{(k)}\|_1 + J^{*(k+1)}(Ax^{(k)} + Bu^{(k)} + Ed^{(k)})$$

$$\mathbb{X}^{(k)} = \{x \in \mathbb{R}^n \; : \forall d \in \mathbb{D} \;\; \exists u \in \mathbb{R}^{n_u} \text{ with } Fx + Gu \le g \text{ and } Ax + Bu + Ev \in \mathbb{X}^{(k+1)}\}.$$

where $J^{*K}(x^{(K)}) = 0$ and $\mathbb{X}^{(K)} = \{x \in \mathbb{R}^n : Fx \le g\}$

Model based predictive control

Robust MPC

Dynamics

$$x(k+1) = Ax(k) + Bu(k) + Ed(k) \qquad d(k) \in \mathbb{D} = \{d : Ld \leq l\}$$

$$J^{*(k)}(x^{(k)}) = \min_{u^{(k)}} J^{(k)}(x^{(k)}, u^{(k)})$$

$$\text{s.t.} \left\{ \begin{array}{l} Fx^{(k)} + Gu^{(k)} \leq g \\ Ax^{(k)} + Bu^{(k)} \in \mathbb{X}^{(k)} \end{array} \right\} \forall d^{(k)} \in \mathbb{D}$$

$$J^{(k)}(x^{(k)}, u^{(k)}) = \max_{d^{(k)} \in \mathbb{D}} \|Qx^{(k)}\|_1 + \|Ru^{(k)}\|_1 + J^{*(k+1)}(Ax^{(k)} + Bu^{(k)} + Ed^{(k)})$$

$$\mathbb{X}^{(k)} = \{x \in \mathbb{R}^n \ : \forall d \in \mathbb{D} \ \ \exists u \in \mathbb{R}^{n_u} \text{ with}$$

$$Fx + Gu \leq g \text{ and } Ax + Bu + Ev \in \mathbb{X}^{(k+1)}\}.$$

where $J^{*K}(x^{(K)}) = 0$ and $\mathbb{X}^{(K)} = \{x \in \mathbb{R}^n : Fx \leq g\}$

Robust MPC

Dynamics

$$x(k+1) = Ax(k) + Bu(k) + Ed(k) \qquad d(k) \in \mathbb{D} = \{d : Ld \leq l\}$$

$$J^{*(k)}(x^{(k)}) = \min_{u^{(k)}} J^{(k)}(x^{(k)}, u^{(k)})$$

$$\text{s.t.} \left\{ \begin{array}{l} Fx^{(k)} + Gu^{(k)} \leq g \\ Ax^{(k)} + Bu^{(k)} \in \mathbb{X}^{(k)} \end{array} \right\} \forall d^{(k)} \in \mathbb{D}$$

$$J^{(k)}(x^{(k)}, u^{(k)}) = \max_{d^{(k)} \in \mathbb{D}} \|Qx^{(k)}\|_1 + \|Ru^{(k)}\|_1 + J^{*(k+1)}(Ax^{(k)} + Bu^{(k)} + Ed^{(k)})$$

$$\mathbb{X}^{(k)} = \{x \in \mathbb{R}^n \; : \forall d \in \mathbb{D} \;\; \exists u \in \mathbb{R}^{n_u} \text{ with}$$

$$Fx + Gu \leq g \text{ and } Ax + Bu + Ev \in \mathbb{X}^{(k+1)} \}.$$

where $J^{*K}(x^{(K)}) = 0$ and $\mathbb{X}^{(K)} = \{x \in \mathbb{R}^n : Fx \leq g\}$

Robust MPC

Dynamics

$$x(k+1) = Ax(k) + Bu(k) + Ed(k) \qquad d(k) \in \mathbb{D} = \{d : Ld \leq l\}$$

$$J^{*(k)}(x^{(k)}) = \min_{u^{(k)}} J^{(k)}(x^{(k)}, u^{(k)})$$

$$\text{s.t.} \left\{ \begin{array}{l} Fx^{(k)} + Gu^{(k)} \leq g \\ Ax^{(k)} + Bu^{(k)} \in \mathbb{X}^{(k)} \end{array} \right\} \forall d^{(k)} \in \mathbb{D}$$

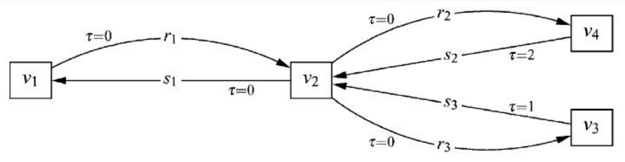
$$J^{(k)}(x^{(k)}, u^{(k)}) = \max_{d^{(k)} \in \mathbb{D}} \|Qx^{(k)}\|_1 + \|Ru^{(k)}\|_1 + J^{*(k+1)}(Ax^{(k)} + Bu^{(k)} + Ed^{(k)})$$

$$\mathbb{X}^{(k)} = \{x \in \mathbb{R}^n \ : \forall d \in \mathbb{D} \ \ \exists u \in \mathbb{R}^{n_u} \text{ with } \\ Fx + Gu \leq g \text{ and } Ax + Bu + Ev \in \mathbb{X}^{(k+1)} \}.$$

where $J^{*K}(x^{(K)}) = 0$ and $\mathbb{X}^{(K)} = \{x \in \mathbb{R}^n : Fx \leq g\}$

Model based predictive control

Robust MPC: Example



Retailer v_2: uncertain demand $d(t) \in [0, 8]$

- Order $u_1(t) \in [0, 6]$ from supplier v_3: cost 4, delay 1
- Order $u_2(t) \in [0, 6]$ from supplier v_4: cost 1, delay 2

Technische Universiteit Eindhoven University of Technology

Robust MPC: Example

$$x(t+1) = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u(t) + \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} d(t).$$

Resulting (dual base stock) policy:

$$u_1^*(x) = \min\{\max\{20 - x_1 - x_2 - x_3 - x_4, 0\}, 4\},$$
$$u_2^*(x) = \max\{16 - x_1 - x_2 - x_3 - x_4, 0\}.$$

TU/e Technische Universiteit Eindhoven University of Technology