Min-Max Model Predictive Control
Implementation Strategies
Daniel R. Ramirez & Eduardo F. Camacho

Outline
- Why Min-Max Model Predictive Control?
- Min-Max Model Predictive Control using open-loop predictions.
- Min-Max Model Predictive Control using closed-loop predictions.

Why Min-Max Model Predictive Control?

Model Predictive control is a success both in the industrial community and academia:
- 4600 Applications (Qin y Badgwell, 2002).
- 67% Refining and petrochemical industry.
- Hundreds of papers published.

Why Min-Max Model Predictive Control?

PROS:
- Multivariable from the beginning.
- Deadtime compensation.
- Optimal design.
- Guaranteed stability (academic version).
- Able to consider constraints.

CONS:
- Computational burden.
- Need of a model.
- Modelling errors (uncertainties, disturbances) can degrade the closed loop performance.

Robust Model Predictive Control:
- Robust stability guaranteed:
  - Keep the state within a region.
  - Robust constraint satisfaction.
- Improve performance in presence of uncertainties and disturbances:
  - Improving robustness.
  - Take into account uncertainties in the design.

Min-Max formulation: worst case optimized design
Min-Max MPC (MMMPC)
Campo y Morari, 1987

Why Min-Max MPC?

Robust MPC:
- Robust stability guaranteed:
  - Keep the state within a region.
  - Robust constraint satisfaction.
- Improve performance in presence of uncertainties and disturbances:
  - Improving robustness.
  - Take into account uncertainties in the design.

Min-Max MPC (MMMPC)
Campo y Morari, 1987

Better performance against uncertainties → More Robustness
Why Min-Max Model Predictive Control?

This increase in robustness also appear in real world applications.

Control Engineering Practice, 2005

More robustness can also help avoiding feasibility problems:
1. Constraint violation due to modelling errors.
2. Unfeasible optimization problem.

Even when there are advantages the number of reported applications is very low.

Computational Burden issues

Desirable properties of an Min-Max MPC:
- Far lower computational burden, complexity growing with prediction horizon at non exponential rate.
- Use efficient numerical algorithms (such as in QP) to compute control signal.
- Flexible implementation, able to change parameters in real-time.
- Able to use constraints.
- Low error if an approximation of the optimizer is used.
- Robust stability guaranteed.

Types of Min-Max MPC

A broad division of Min-Max MPC can be established between those which use open-loop predictions and those with closed-loop predictions (note that despite its type all of them are feedback controllers).

- Open-loop prediction based controllers are the most mature algorithms dating back to 1987 (Campo and Morari).
- They suffer from conservatism in the predicted evolution of the process that can lead to poor performance or feasibility problems. However, properly tuned they work well as proved later.
- Although its computational burden grows exponentially with certain parameters, there exists techniques to implement it in a wide class of systems.

On the other hand, closed-loop prediction based controllers have lower conservation and better feasibility properties, because they take into account the fact that the control law is applied in a feedback manner.

- The price to be paid is an even greater computational burden and fewer implementation options (in fact there is no any single reported application to a real process).
- These controllers date from 1997 (Lee and Yu) and 1998 (Scokaert et al) and usually rely in recursive min-max problems or the optimization of a set control policies instead of a sequence of control signal values.
- Finally there is a compromise between these two types of controllers called the semi-feedback approach, that can be used with open-loop prediction controllers to add a certain degree of closed-loop behavior in the predictions.
Open loop vs close loop prediction

Predicted state: due to the uncertainty the state $x_k$ lies in the Reachable Set $W_k$, $k \in N_k$.

$x_k = x_0 + N + w_0 = w_0 + w_0 \Rightarrow x_k \in W_0 \subseteq [-1,1]$.

$x_1 = x_1 + w_1 \in (N + w_1) \subseteq [-1,1] \Rightarrow x_1 \in [-1,1]$.

$x_2 = x_2 + w_2 \in (N + w_2) \subseteq [-3,3]$.

$x_3 = x_3 + w_3 \in (N + w_3) \subseteq [-k, k]$.

$x_N = x_N + w_N \in (N + w_N) \subseteq [-N, N]$.

If $N > 0$, Unfeasible Problem!

Min-Max Model Predictive Control based on Open-Loop Predictions

We use the knowledge of $x_k$ at time $k$ through a feedback:

$u_k = K_k x_k$,

with $K_k \in \mathbb{R}$, for all $k \in N_k$.

Objective of the Closed-Loop MPC

$\min \max J(x_k, K, u)$

with constraints

$x_0 \in [-1,1]$,

$u_k = K_k x_k \in U = \mathbb{R}$

where $K = \{ K_0, K_1, \ldots, K_{N-1} \}$ and $w = \{ w_0, w_1, \ldots, w_{N-1} \}$.

MMMPC using Open-Loop Predictions

- Robust Model Predictive Control
- Explicit implementation
- Implementation using a nonlinear bound
- Implementation using a QP based bound
- Applications and Examples

Predictive Control

At each sampling time $t$:

- Get new measures $y(t)$
- Estimate $x(t)$
- Solve a finite time optimal control problem to obtain $u^*(t)$
- Apply the first component of $u^*(t)$
Robust Predictive Control

Based on models that take into account uncertainties:

\[ y(t+1) = f(y(t), \ldots, y(t-n)), u(t), \ldots, u(t-n)) + \theta(t) \]

\[-\kappa \leq \theta(t) \leq \kappa\]

**Output Predictions**

**Variables**

**Manipulated**

\[ u(t+k) \]

\[ y(t+k) \]

**Based on models that take into account uncertainties:**

**UNCERTAIN PREDICTIONS**

Robust Model Predictive Control

System model:

\[
\begin{align*}
x(t+1) &= Ax(t) + Bu(t) + Du(t+1) \\
y(t) &= Cx(t)
\end{align*}
\]

\[ |\theta(t)|_{w} \leq \varepsilon \]

**Bounded additive uncertainties**

Two strategies to consider \( u(t) \):

- Open-loop predictions: \( w(t), w(t+1), w(t+2), \ldots \)
- Semi-feedback predictions:

\[
w(t) = -Kx(t) + w(t)
\]

\[
a(t+1) = A_{x(t)} + Bu(t) + Du(t+1)
\]

**Computed by the controller**

Robust MPC

The inner loop pre-stabilizes the nominal system

Min-max MPC

The prediction equation: \( y = G_{u}u + G_{w}w + f \)

Defining \( g(u, \theta) = (y - w) \), the control problem can be expressed as

\[
\min_{u} \max_{\theta \in \Theta} \max_{i = 1 \cdots n} \left| g_{i}(u, \theta) \right|
\]

Define \( \mu^{*}(u) \) as

\[
\mu^{*}(u) = \max_{\theta \in \Theta} \max_{i = 1 \cdots n} \left| g_{i}(u, \theta) \right|
\]

If there is any positive real value \( \mu \) satisfying \( -\mu \leq g(u, \theta) \leq \mu \), \( \forall \theta \in \Theta \) and for \( i = 1 \cdots n \times N \) it is clear that \( \mu \) is an upper bound of \( \mu^{*}(u) \).

Min-max MPC (\( \infty \)-\( \infty \) norm)

- Campo and showed that by using an \( \infty \)-\( \infty \) norm the min–max problem reduces to a linear programming problem.
- Although the algorithm was developed for the truncated impulse response, it can easily be extended to the other models used.
- The objective function is now described as

\[
J(u, \theta) = \max_{j=1 \cdots N} \left| g_{j}(u(t+j)) - w(t) \right|_{\infty} = \max_{j=1 \cdots N} \max_{i=1 \cdots n} \left| g_{j}(u(t+j)) - w_{i}(t) \right|
\]

Min-max MPC (\( \infty \)-\( \infty \) norm)

Find the smallest upper bound \( \mu \) and some \( u \in U \) for all \( \theta \in \Theta \).

When constraints on the controlled variables \( (y, y') \) are taken into account, the problem can be expressed as

\[
\min_{\mu \in \mathbb{R}} \mu
\]

subject to

\[
\begin{align*}
-\mu \leq g_{j}(u, \theta) - \mu & \leq \mu \quad \text{for } i = 1 \cdots n \times N \\
y_{j} - w_{i} - g_{j}(u, \theta) & \leq y_{j} - w_{i} \quad \forall \theta \in \Theta
\end{align*}
\]
Min-max MPC ($\infty - \infty$ norm)

If $\rho(u, \theta)$ is an affine function of $\theta$, $\forall u \in U$, the maximum and minimum of $\rho(u, \theta)$ can be obtained at one of the extreme points of $\Theta$.

Let $E$ be the set formed by the $2^n \times N$ vertices of $\Theta$.

If constraints are satisfied for every point of $E$ they will also be satisfied for every point of $\Theta$.

Thus the infinite, and continuous, constraints can be replaced by a finite number of constraints.

Min-max MPC (1-norm)

$J(u, \theta) = \sum_{j=1}^{N_u} \sum_{j=1}^{N_u} |u_j(t+j) - u_j(t+j)| + \lambda \sum_{j=1}^{N_u} \sum_{j=1}^{N_u} |\Delta u_j(t+j-1)|$

If a series of $\mu_i \geq 0$ and $\beta_i \geq 0$ such that for all $\theta \in \Theta$,

$-\mu_i \leq \Delta u_i(t+j) \leq \mu_i$

$-\beta_i \leq \Delta u_i(t+j-1) \leq \beta_i$

then $\gamma$ is an upper bound of $\mu^*(u) = \max_{\theta \in \Theta} \sum_{j=1}^{N_u} \sum_{j=1}^{N_u} |u_j(t+j) - u_j(t+j)| + \lambda \sum_{j=1}^{N_u} \sum_{j=1}^{N_u} |\Delta u_j(t+j-1)|$

Min-max MPC (quadratic cost)

Cost Function:

$V(x, u, \theta) = \sum_{j=0}^{N} \pi(t+j) \Pi Q(t+j) + \sum_{j=0}^{N} \pi(t+j) \Pi R(t+j) + \sum_{j=0}^{N} \pi(t+j) \Pi P(t+j)$

Constraints:

$F_o(t+j) + F_u(t+j) \leq 0$

$0, \ldots, N, \forall \theta \in \Theta$

Min-max MPC (quadratic cost)

The Min-Max Strategy:

$V^*(x, u) = \max_{\theta \in \Theta} V(x, u, \theta)$

The maximum is attained at least at one of the vertices of $\Theta$.

Min-max MPC (quadratic cost)

Cost Function:

$V(x, u, \theta) = \sum_{j=0}^{N} \pi(t+j) \Pi Q(t+j) + \sum_{j=0}^{N} \pi(t+j) \Pi R(t+j) + \sum_{j=0}^{N} \pi(t+j) \Pi P(t+j)$

Constraints:

$F_o(t+j) + F_u(t+j) \leq 0$

$0, \ldots, N, \forall \theta \in \Theta$
Min-max MPC (quadratic cost)

Thus, the max function can be computed as:

\[ V^*(x, u) = \max_{\theta \in WSF(\Theta)} V(x, u, \theta) \]

Number of vertices: \( 2^{N \cdot \text{dim} \Theta} \)

Exponential complexity

NP hard problem

As a result, the min-max problem is of the NP hard kind and in general it cannot be solved in real time except for slow processes and/or small horizons.

Real time implementation of Min-Max MPC

How to get an Min-Max MPC that can be computed in real time ??

Some suggestions:

- Use an explicit implementation, in which the min-max problem is precomputed for feasible states.
- Solve an equivalent reduced min-max problem.
- Substitute the worst case cost for a low computational burden close approximation.

MMMPC using Open-Loop Predictions

- Robust Model Predictive Control
- Explicit implementation
- Implementation using a nonlinear bound
- Implementation using a QP based bound
- Applications and Examples

Explicit implementation of Min-Max MPC

An alternative: Multi parametric programming

Explicit form control laws:

\[ u(e) = \begin{cases} 
F_x + G_1 & \text{if } H_x \leq K_1 \\
F_x + G_2 & \text{if } H_x \leq K_2 \\
\vdots & \text{if } H_x \leq K_n 
\end{cases} \]

Multi-parametric Programming

Successfully applied to MPC:

<table>
<thead>
<tr>
<th></th>
<th>Nominal</th>
<th>Min Max Open Loop predictions</th>
<th>Min Max Closed Loop predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic norms</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>LP based norms</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

(Bemporad et.al. AUT 02, Bemporad et.al. TAC 03)
(Rakovic et al. AUT 04, Wen et.al. TAC 04)

Approximate closed loop explicit solutions

Geometrical Approach

- Based on properties of cost function and the concept of active vertices (i.e., the vertices in which the worst case is attained at the solution).
- It allows:
  - To prove that the control law is PWA on the states.
  - To find an explicit solution using a constructive algorithm.

Works well, but... difficult to understand, complex specially with constraints, heuristic if need to be efficient.
Min-max as a Quadratic Program

The min-max problem can be rewritten as:

\[
J^*(x) = \min_{V} \left( V(x, v, 0) + \max_{w \in \text{vert}(W_N)} V(x, w) - V(x, 0) \right),
\]

s.t. \( G_v v + G_x x < \tilde{\eta} \)

Taking into account the cost definition:

\[
V(x, v, w) - V(x, v, 0) = w^T R_1^w H_2^w w + 2w^T R_1^w (H_2 x + H_3 v)
\]

Quadratic dependence on \( w \)
Affine dependence on \( x \) and \( v \)

Min-max as a Quadratic Program

Min max problem:

\[
J^*(x) = \min_{V} \left( V(x, v, 0) + \max_{w \in \text{vert}(W_N)} V(x, w) - V(x, 0) \right),
\]

s.t. \( G_v v + G_x x < \tilde{\eta} \)

Epigraph approach:

\[
J^*(x) = \min_{V} \left( V(x, v, 0) + \gamma \right),
\]

s.t. \( G_v v + G_x x \leq g \)
\( \gamma \geq V(x, v, w) - V(x, v, 0), \forall w \in \text{vert}(W_N) \)

Example

Consider the double integrator with bounded additive uncertainties:

\[
A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}
\]

The uncertainty satisfies: \( |w_u|_{\infty} \leq 1 \)

The state and control input are constrained:

\( 10^2 \leq x_k \leq 10, -1 \leq u_k \leq 1 \)

The control performance objectives are described by

\[
P = Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 10
\]

A linear feedback law is considered: \( K = [-0.2054, -0.7835] \).
Example

Set of active vertices is very small

Feedback PT-326

• Scaled laboratory process
• 2nd order system
• Fast dynamics
  \[ Ts = 0.4 \text{ s.} \]

(Explicit solution)

Results of the MMMPC

Different controllers

Applied with success

(Muñoz de la Peña et al. CEP 05)

Comparisons

Different positions of the inlet throttle from 20° to 100°

Explicit implementation: Pros and Cons

• Possibility to implement explicit min-max MPC control laws, using standard multi-parametric techniques.
• Well suited for fast systems and embedded applications.
• But...
  ☐ The exponential number of constraints limits the value of the prediction horizon.
  ☐ The implementation is not flexible, the user cannot change parameters without recomputing the controller.

MM MPC using Open-Loop Predictions

☐ Robust Model Predictive Control
☐ Explicit implementation
☐ Implementation using a nonlinear bound
☐ Implementation using a QP based bound
☐ Applications and Examples
Using an upper bound of the worst case cost

A computationally cheaper upper bound is minimized instead of the worst case cost:

\[ u^*(x) = \arg \min_{u \in U} \max_{s \in S} \left\{ D^*(u, s) \right\} \]

\[ D^*(u, s) \leq d^* \]

Previous results:

- LMI methods Kothare et al. 96, Lofberg 03
- BQP Algorithms Alamo et al. 04

Strategy based in a nonlinear bound, Ramirez et al., JPC 2006
Strategy based in QP problems, Alamo et al., Automatica 2007

Strategy based on a nonlinear bound

Main contribution:

New upper bound of the maximization problem not based on LMIs or complex algorithms.

\[ \sigma_u(u, x) \geq J^*(u, x) \]

Main Properties:

- Low computational burden.
- Based on matrix computations.
- Close to the worst case cost.

Computing the nonlinear upper bound

The quadratic maximization problem can be rewritten as:

\[ \pi^*(x, u) = \max_{\epsilon \in \{1, \ldots, n\}} \theta^T \gamma + 2\theta^T \psi_n(x, u) + r(u, x) \]

Assume that \( \epsilon = 1 \), then:

Binary optimization problem

\[ \pi^*(x, u) = \max_{\epsilon \in \{1, \ldots, n\}} \theta^T \gamma + 2\theta^T \psi_n(x, u) + r(u, x) \]

Note that matrix \( H \) depends on \((x, u)\)

Computing the nonlinear upper bound

The idea to obtain the bound is:

If \( T \) is a diagonal positive definite matrix such that:

\[ T \geq H \]

then:

\[ \pi^*(x, u) \leq \max_{\epsilon \in \{1, \ldots, n\}} z^T \gamma z = \text{trace}(\gamma) \]

Computation of \( T \):

LMI or Proposed method based on simple matrix computations

Computing the nonlinear upper bound

\( T \) will be obtained adding to \( H \) a series of matrices of the form:

\[ T = H + v_1 v_1^T + v_2 v_2^T + \ldots + v_{n-1} v_{n-1}^T \]

The \( v_i \) will be computed in such a way that allow to obtain a close bound with low complexity.

\[ \sigma_u(u, x) = \text{trace}(T(u, x)) \]

Computing the nonlinear upper bound

Vectors \( v_i \) are computed so that matrix \( H \) is partially diagonalized:

\[ \begin{bmatrix} a & b^T \\ b & H_r \end{bmatrix} + vv^T = \begin{bmatrix} d & 0 \\ 0 & \tilde{H_r} \end{bmatrix} \]

If \( v = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \), Free parameter

\[ d = a + \alpha^2 \]

This idea is applied recursively

\[ \tilde{H_r} = H_r + \frac{\beta}{\alpha^2} \]
Computing the nonlinear upper bound

The error introduced at each diagonalization step must be minimum:

\[ \text{Error} = 3^T \theta e^T z = 3 \left[ \frac{\alpha}{b} \frac{\alpha}{a} \right] \left[ \alpha - \frac{\alpha}{b} \right] \]

\[ \min \text{Error} \]

\[ \alpha = \sqrt{||b||_1} \]

Nonlinear upper bound algorithm

Let \( T = H \in \mathbb{R}^{n \times n} \)

For \( k=1 \) to \( n-1 \)

Let \( H_{k+1} = [T_{ij}] \) for \( i,j = k,...,n \)

Compute \( \alpha \) for \( H_{k+1} \)

Form \( \psi_k = \left[ \alpha - \frac{\alpha}{b} \right] \)

Form \( \psi_k^T = \left[ 0 ... 0 \psi_k^T \right] \in \mathbb{R}^n \)

\[ T = T + \psi_k \psi_k^T \]

End For

When the procedure is over, the bound is computed as \( \sigma_u(H) = \text{trace}(T) \)

Computing the control signal

At each sampling time, solve:

\[ u^*(x) = \arg \min_n \min_{\alpha} \left( r(n,x) \right) \]

and apply the first component of \( u^*(x) \) using a receding horizon strategy.

If at any step of the diagonalization procedure, all the elements of \( H_{k+1} \) are \( \geq 0 \) \( \rightarrow \) the best bound will be:

\[ \sum_{i=1}^{n} \psi_i^T \]

No need to continue the computation loop

Accuracy of the nonlinear upper bound

It will be compared against the LMI bound:

\[ \sigma^* = \min_n \text{trace}(T) \]

\[ T \geq H \]

\[ T \text{ diagonal} \]

Deviation for a set of random matrices of the form \( H=H_0'H_0 \).

Typical matrices in MPC problems are outside this zone, because the mean of its elements are different from zero

Typical matrices in MPC problems are outside this zone, because the mean of its elements are different from zero

Computational burden of the bound

Relative speed up from the LMI bound:

Solver LMI by F. Rendl

http://www.math.uni-klu.ac.at/or/Software.

MMMPC using Open-Loop Predictions

- Robust Model Predictive Control
- Explicit implementation
- Implementation using a nonlinear bound
- Implementation using a QP based bound
- Applications and Examples
QP based strategy

- It has two desirable properties:
  - Robust stability guaranteed.
  - Implementation based on an optimization problem similar to that of conventional MPC \(\rightarrow\) Quadratic Programming (QP).

- Outline of the strategy:
  1. Obtain an initial estimation of the solution of the min-max problem.
  2. Obtain a quadratic function that bounds the worst case cost.
  3. Use a diagonalization procedure to obtain the quadratic function.
  4. Compute the control signal as the minimizer of the quadratic function.

Robust stability guaranteed, but requires semi-feedback for non open-loop stable systems:

\[
\begin{align*}
\text{Stabilizing design} & \\
\text{QP problems based strategy} & \\
\end{align*}
\]

Stabilizing design

- Conditions on the terminal region

\[
\begin{align*}
C1: \ x \in \Omega & \rightarrow \ A_{CL}x + D_1l \in \Omega, \forall l \\
C2: \ x \in \Omega & \rightarrow \ F_1(-x) + F_2x \leq g
\end{align*}
\]

- Admissible robust invariant set

\[
\begin{align*}
\text{If there exists a solution for } x(k) & \text{ then exists solution for } x(k+1) \\
\text{Convergence to the origin} & \text{ (Optimal cost } \rightarrow \text{ Lyapunov function)}
\end{align*}
\]

QP problems based strategy

- Step 1: Initial estimation of the solution of the min-max problem

\[
\begin{align*}
V^*(x,v) = V(x,v,0) + \max_{\omega_m(v)} \delta^T \theta + 2\delta^T g(x,v)
\end{align*}
\]

- An easily computable (although not very good) upper bound is:

\[
\begin{align*}
V^*(x,v) = V(x,v,0) + \|\Pi_{x=0}\|^2 + 2\epsilon_1|v|_2 x_2 \geq V(x,v)
\end{align*}
\]

Step 2: Computing the quadratic function

- Step 2.1: Compute the parameters \(a_k\) for the initial estimation

\[
\begin{align*}
\Pi(x,v) = \max_{\omega_m(v)} \theta^T \Pi \left( \nu, \delta, \Gamma \right) \theta = \max_{\omega_m(v)} \Pi M(\delta, x,v)
\end{align*}
\]

- Take into account that if it is assumed \(\epsilon = 1\):

\[
\begin{align*}
a(x(v),x) = |a_1(x(v))|, \ldots, a_n(x(v))^{\Delta T}
\end{align*}
\]

- The resulting diagonal matrix, denoted by \(\Sigma(v)\) verifies:

\[
\begin{align*}
V^*(x,v) \leq \max_{\omega_m(v)} \Pi M(\delta, x,v)
\end{align*}
\]
Step 3: Computing the signal control

\[ V^*(x, v) = \text{trace } (F^T v) \]

1. \( V^*(x, v) \) is a quadratic function of \( v \)
2. \( V^*(x, v) \leq V^*(x, v') \)

- The control signal will be computed solving:
  \[ \hat{v}^*(x) = \arg\min_{\gamma} V^*(x, \gamma) \]
  s.a. \( J_\gamma + F_\gamma x \leq d_\gamma \)

- The suboptimality is bounded:
  \[ V^*(x, v) \leq V^*(x, 0) \leq V^*(x, \hat{v}^*) + \sigma^2 \]
  \[ \sigma = \|H\|_1 \]

Stability

- The state is always steered into \( \Phi_\varepsilon \), but it can be escape from it.
- The set to which the state can evolve from \( \Phi_\varepsilon \) is:

\[ \Omega_\beta = \left\{ x \in \mathbb{R}^{n_{\text{states}}}: J(x) < \beta \right\} \]

\[ \beta = \min_{\gamma \in \Phi_\varepsilon} J(x) + (\gamma + \varepsilon)^2 \]

Stability

- The proposed control law guarantees the closed loop robust stability.
- Outline of the proof:
  - Let \( J(x(t)) \) be the optimal cost of the original min-max problem.
  - It can be proved that using the proposed control law there exists \( \gamma > 0 \) such that:
    \[ J(x(t+1)) - J(x(t)) \leq \sigma \gamma \]
  - This implies that:
    \[ J(x(t+1)) - J(x(t)) \leq \epsilon \]
    \[ J(x(t+1)) - J(x(t)) \leq \epsilon \]
  - That is, from any \( x(t) \) the state evolves into:
    \[ \Phi_\beta = \left\{ x \in \mathbb{R}^{n_{\text{states}}}: \text{stable and } x(t) \leq (\gamma + \varepsilon)^2 \right\} \]

MM MPC using Open-Loop Predictions

- Robust Model Predictive Control
- Explicit implementation
- Implementation using a nonlinear bound
- Implementation using a QP based bound
- Applications and Examples

A simulated example

Two tank process:
- Tanks sections: 3 m² and 2 m².
- \( a1 = a2 = 0.5 \text{ m}^2\text{min}^{-1} \)
- 40% of the liquid is pumped back to tank 1.

\[ \dot{x} = \begin{bmatrix} 0.5 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} x + \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} x \]

A simulated example

Controller based on the nonlinear bound
- \( \varepsilon = 0.02 \)
- \( N=15 \)
- \( N_u=10 \)
- \( 2^m \) vertices

- \( x_1 < x_{\text{max}} < 0.5 \)
- \( x_2 < x_{\text{min}} < 0.5 \)
- \( 0.01 \leq J_{\text{min}} \leq 0.02 \)
**Computational burden**

Average, max and min speed-up over the exact MMMPC.

**Accuracy of the nonlinear bound**

Average, max and min speed-up over the minimization of the LMI bound.

**Accuracy of the nonlinear bound**

Desviation from the exact optimal cost and from the optimal cost obtained using the LMI bound.

**A simulated example**

Optimal cost of the exact MMMPC and the QP based controller (N=7, Nu=7).

Optimal cost of the exact MMMPC when it is used both the exact solution and that of the QP based controller, that is \( y^*(x, t), y^x(x, t) \).

**A simulated example**

Average, max and min speed-up for the QP based controller over the exact MMMPC.
Application to a pilot plant

- Tank with a 14.4 kW heater.
- Recirculation through a heat exchanger.
- Industrial instrumentation.
- Connected to Simatic-IT.
- It emulates a CSTR using the heater to simulate the heat of an exothermic reaction.

Model by Santos et al., 2002

Comparison for a set point tracking experiment using:
- MPC.
- MMMPC based on the nonlinear bound.
- MMMPC based on QP problems.

Disturbances in the inlet flow (25%)
In order to evaluate reference tracking, the state vector is augmented:

\[ z_k = \begin{bmatrix} y_{k+d-1} & y_{k+d-2} & \Delta u_{k-1} & \Delta u_{k-2} & \Delta u_{k-3} & r_k \end{bmatrix}^T \]

Fixed reference over the prediction horizon

\[ (y_k - r_k)^2 = z_k^T Q z_k \]

Quadratic criterion

Best model:

\[ A(z^{-1}) = 1 - 0.5510z^{-1} - 0.4072z^{-2} \]
\[ B(z^{-1}) = 0.0090 + 0.0127z^{-1} + 0.0105z^{-2} \]

Delay = 4

Error bound = 0.05

**CARIMA**

\[ \Delta A(z^{-1}) y_k = z^{-d} B(z^{-1}) \Delta u_{k-1-d} + \theta_k \]
Results of the MMMPC

Different controllers

"Well" tuned MMMPC

Aggressive control law

Open loop predictions

Linear feedback:

\[ \Delta u_k = K z_k + v_k \]

MMMPC with Linear Feedback

Feedback on the predictions without complexity increase (Bemporad et al., 1998)

MMMPC with Linear Feedback

MMMPC with Linear Feedback

Comparison of the MMMPC with linear feedback

Different controllers

"Well" tuned MMMPC

"Softer" control law

Feedback in the predictions

Comparisons

Different positions of the inlet throttle from 20° to 100°

MMMPC

Open loop predictions

MMMPC with linear feedback

Control Input

Comparisons

 MMMPC with Linear Feedback
Min-Max Model Predictive Control using Closed-Loop Predictions

Feedback Min-Max MPC

\[ F(x) = \min_{u} \max_{w} J(x, u, w) \]
subject to:

Feedback Min-Max MPC

\[ F(x) = \min_{u} \max_{w} \left( l(x_0, u_0) + \max_{w'} \min_{u'} \left( l(x_1, u_1) + \ldots + \max_{w_{N-1}} \left( l(x_{N-1}, u_{N-1}) + F(x_N) \right) \right) \right) \]
subject to:

How to implement it??

- Dynamic Programming can be used to state the closed-loop min-max MPC.
- The problem can be expressed as the recursive problem.

\[ f_r (x(t)) \preceq \min_{u(t)} l_r (x(t), u(t)) \]
subject to:

The Closed-Loop Min-Max MPC

- Where \( X(t+1) \) is the region where function \( f_{r+1} (x(t+1)) \) is defined.
The Closed-Loop Min-Max MPC (2)

- \( x(t+1) = A(\Theta(t))x(t) + B(\Theta(t))u(t) + E(\Theta(t)), \)
- with \( A(\Theta(t)) = A_0 + \sum_{i=1}^{n_{\Theta}} A_i \Theta_i(t), \)
- \( B(\Theta(t)) = B_0 + \sum_{i=1}^{n_{\Theta}} B_i \Theta_i(t), \)
- \( E(\Theta(t)) = E_0 + \sum_{i=1}^{n_{\Theta}} E_i \Theta_i(t). \)
- Where \( \Theta_i(t) \) denotes the \( i \)th component of the uncertainty vector \( \Theta(t). \)
- The stage cost of the objective function defined as:
  \( L(x(t+j), u(t+j); ||Qx(t+j)||^p + ||Ru(t+j)||^p) \)
- With the terminal cost defined as
  \( J^*(x(t)) = \min_{u(t)} J(x(t), u) \)
- s.t. \( Ru(t) \geq r + Sx(t) \)
- The solution is a piecewise affine function of the state.
  (Bemporad et. al. 2003)

The Closed-Loop Min-Max MPC (3)

- The demonstration is based on the following
  1. If \( J(x(t), u) \) is a convex piecewise affine function
     (i.e., \( J(x(t), u) = \max_{i=1,\ldots,s} \{L_i + H_i x(t) + k_i\} \))
     the problem:
       \( J^*(x(t)) = \min_{u(t)} J(x(t), u) \)
       s.t. \( Ru(t) \geq r + Sx(t) \)
   is equivalent to the following mp-LP problem:
     \( \min_{u(t)} \epsilon \)
     s.t. \( L_i(\theta_i)u + H_i(\theta_i)x(t) + k_i(\theta_i) \leq \epsilon \)

The Closed-Loop Min-Max MPC (4)

- 2. If \( f(u, x(t), \Theta) \) and \( g(u, x(t), \Theta) \) are convex
   functions in \( \Theta \) for all \( (u, x(t)) \) with \( \Theta \in \Theta \), where \( \Theta \)
   is a polyhedron with vertices \( \Theta_i (i=1,\ldots,N_{\Theta}) \)
   then the problem
     \( J^*(x(t)) = \min_{u(t)} \max_{\Theta_i} J(x(t), u, \Theta) \)
     s.t. \( g(u(t), x(t), \Theta_i) \leq 0 \)
   is equivalent to the problem:
     \( \min_{u(t), \epsilon} \epsilon \)
     s.t. \( J(x(t), u(t), \Theta_i) \leq \epsilon \)

The Closed-Loop Min-Max MPC (5)

- 3. If \( f(u, x(t), \Theta) \) is convex and piecewise in \( x(t) \) \( u \)
   (i.e., \( f(u, x(t), \Theta) = L_i(\theta_i)u + H_i(\theta_i)x(t) + k_i(\theta_i) \))
   and \( g(u, x(t), \Theta) \) is affine in \( x(t) \) and \( u \) (i.e., \( g(u, x(t), \Theta) = L_g(\theta_i)u + H_g(\theta_i)x(t) + k_g(\theta_i) \))
   4. and \( L_i, H_i, k_i \) convex functions,
   5. then the min-max problem is equivalent to the problem:
     \( \min_{u(t)} \epsilon \)
     s.t. \( L_i(\theta_i)u + H_i(\theta_i)x(t) + k_i(\theta_i) \leq \epsilon \)
     \( L_g(\theta_i)u + H_g(\theta_i)x(t) + k_g(\theta_i) \leq 0 \)

The Closed-Loop Min-Max MPC (6)

- Let us now consider the first step of the
  dynamic programming problem (11.36)
  with
  \( L(x, u) = ||Qx(t+j)||^p + ||Ru(t+j)||^p \)
- The terminal cost
  \( J^*(x(t+N)) = ||Px(t+N)||^p \)
- and the linear system
  \( x(t+1) = A(\theta(t))x(t) + B(\theta(t))u(t) + E(\theta(t)). \)
• $J_{w_j}$ is the solution of the mp-LP problem which results in a piecewise affine function of $x (N-j)$.
• The corresponding control signal $u^* (N-1)$ is also a continuous piecewise affine function of the state.
• The feasible set $X (N-1)$ is a convex polyhedron.
• The same recursively for $j = N-2, N-3, ... 0$.
• We can conclude that $u^* (t)$ is a piecewise affine function of the state $x (t)$.
• Notice that the argument does not hold when the objective function is quadratic as function (11.42) would not be piecewise affine convex with respect to the maximization variables $\theta (N-1)$.

The Closed-Loop Min-Max MPC (8)

The number of regions grows exponentially with the horizon, dimension of state vector and number of vertices of uncertain polytope.

The state vector dimension which can be very high for processes with long dead times. (as can be found frequently in industry)

In the case of a dead time of $d$ sampling instants, an augmented state vector

$x_a (t) = [x (t) u (t-1)^T ... u(t-d)^T]$.

Fast Implementation of MPC and Dead Time Considerations

The number of regions grows exponentially with the horizon, dimension of state vector and number of vertices of uncertain polytope.

The state vector dimension which can be very high for processes with long dead times. (as can be found frequently in industry)

In the case of a dead time of $d$ sampling instants, an augmented state vector

$x_a (t) = [x (t) u (t-1)^T ... u(t-d)^T]$.

To overcome these problems.

- Use the predicted state $\hat{x}(t + d | t)$ to compute $f_{MPC}(\hat{z}(t + d | t))$ instead of $f_{MPC}(x_a (t))$.

Consider a process modelled by the reaction curve method with a dead time equal to its time constant. If the sampling time chosen is one-tenth of the time constant, then $\dim(x_a (t)) = 11$ while $\dim(x (t)) = 1$.

Multi-parametric Convex Programming

Feedback MPC can be often solved by convex optimization:

$$W^*(z) = \min_{z} W(z, x)$$
$$\text{s.t. } g_i(z, x) < 0, i = 1, \ldots, p$$

$W(z, x)$: Convex objective function
$g_i(z, x)$: Convex constraints
$z$: Optimization variables, $x$: Parameters

Approximate multi-parametric convex programming solver:
A. Bemporad and C. Filippi, Approximate Multi-parametric Convex Programming, CDC 2003

Multi-parametric Convex Programming

Properties:

- Piecewise affine (sub-)optimal solution
- Constraint satisfaction
- (sub-)optimality with guaranteed error bound
- Tree structure solution
Multi-parametric Convex Programming

Kothare’s controller properties:

1. The control law robustly regulates the system to the origin while assuring robust constraint satisfaction.

\[
V^*(x_{k+1}) - V^*(x_k) \leq \frac{1}{2} x_k^T Q x_k
\]

2. Approximate control law:

The control law ultimately bounds the system in a region that contains the origin while assuring robust constraint satisfaction.

\[
V^*(x_{k+1}) - V^*(x_k) \leq \frac{1}{2} x_k^T Q x_k + \epsilon
\]

Very efficient implementation

(Kothare et al., Automatica 1996)

(Munoz de la Peña et al., TAC 2005)

Decomposition Algorithm

Feedback Min-Max MPC

\[
F^*(x) = \min_{x_0} \max_{u_0} \left\{ L(x_0, u_0) + \max_{x_1} \left( L(x_1, u_1) + \ldots + \max_{u_{N-1}} L(x_{N-1}, u_{N-1}) + F(x_N) \right) \right\}
\]

subject to:

\[
x_j \in X, j = 0, \ldots, N - 1.
\]

\[
x_N \in \Omega,
\]

\[
u_j \in U, j = 0, \ldots, N - 1.
\]

Linear systems with bounded uncertainties

Additive uncertainties

Polytopic uncertainties

Decomposition Algorithm

Recursive application of mlpLP solvers

Explicit form control laws:

\[
F^*(x) = \min_{x_0} \max_{u_0} \left\{ L(x_0, u_0) + \max_{x_1} \left( L(x_1, u_1) + \ldots + \max_{u_{N-1}} L(x_{N-1}, u_{N-1}) + F(x_N) \right) \right\}
\]

\[
subject to:
\]

\[
x_j \in X, j = 0, \ldots, N - 1.
\]

\[
x_N \in \Omega,
\]

\[
u_j \in U, j = 0, \ldots, N - 1.
\]

Nodes of the tree grow exponentially with N

Cost functions based on a LP problem:

No similar result

Cost functions based on a QP problem:

Nodes of the tree grow exponentially with N
Decomposition Algorithm

Cost functions based on a LP problem:
Equivalent problem. Multi-Stage Min Max Linear Problem.

\[
V^n_j(s_{0:j}) = \min_{z_j} c_j^T z_j + \max_{\nu_{j+1}} V^{n+1}_j(s_{j+1}),
\]

s.t. \[ W_{j+1} = h_j - A_j s_{j+1} \nu_{j+1} \nu_{j+1} > 0. \]

Each node:
Cost-to-go function \( V^n \)
Set of variables \( z_j \)
Depends on uncertainty

Cost functions based on a QP problem:
Equivalent problem. Multi-Stage Min Max Quadratic Problem.

\[
V^n_j(s_{0:j}) = \min_{z_j} c_j^T z_j + \max_{\nu_{j+1}} V^{n+1}_j(s_{j+1}),
\]

s.t. \[ W_{j+1} = h_j - A_j s_{j+1} \nu_{j+1} \nu_{j+1} > 0. \]

Main contribution
It allows on-line implementation of feedback min-max MPC.
(For a broader family of systems)

Algorithm that solves
Based on the ideas introduced by Benders in 1962
MILP problems, LP problems, SP problems

Decomposition Algorithm

Function \( \max_{j, k} V^n_j(s) \) is PWA
Algorithm converges to optimum

Decomposition Algorithm

Function \( \max_{j, k} V^n_j(s) \) is PWQ
Bound on the error is a parameter

Decomposition Algorithm

MMMPC using Closed-Loop Predictions

- Feedback Min-Max MPC
- Approximate Multi-parametric Programming
- Decomposition algorithm
- Example
Numerical Examples

Quadruple tank process

Model of a real plant

\[
\begin{pmatrix}
0.8541 & 0 & 0.1032 & 0 \\
0 & 0.9100 & 0 & 0.0503 \\
0 & 0 & 0.8843 & 0 \\
0 & 0 & 0 & 0.9473
\end{pmatrix}
\]

\[n_k + 1 \]

\[+ \begin{pmatrix}
0.0129 & 0.0015 \\
0.0098 & 0.0177 \\
0.0316 & 0
\end{pmatrix} u_k + \begin{pmatrix}
1 & 0 \\
0 & 1 \\
-1 & 0
\end{pmatrix} u_{k-1}
\]

\[n_k = 4\]

Numerical Examples

Computation times:

<table>
<thead>
<tr>
<th>N</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.094</td>
<td>0.496</td>
<td>1.76</td>
<td>6.93</td>
<td>27.4</td>
<td>+2min</td>
</tr>
</tbody>
</table>

It can be implemented up to N=5

More than 2000 nodes

Only algorithm available in the literature.

Problem: Complexity still grows exponentially with N.

Good performance... N=10

Proposed solution: RECURSE HORIZON - Still an open problem

Finally, some papers


