Introduction to Nonlinear Control

Stability, control design, and estimation

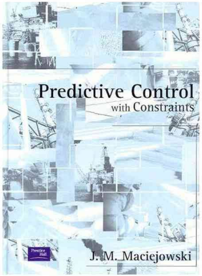
Philipp Braun & Christopher M. Kellett
School of Engineering,
Australian National University, Canberra, Australia

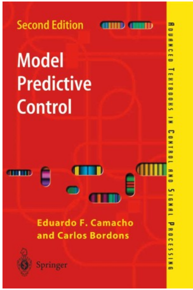
Part II:

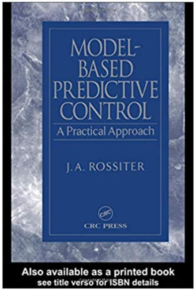
Chapter 15: Model Predictive Control

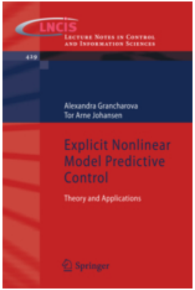
Australian National University

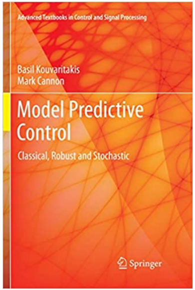
Model Predictive Control

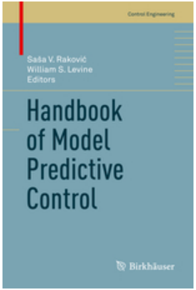


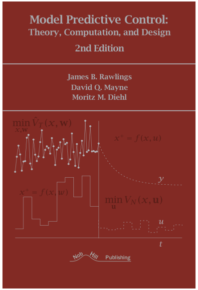


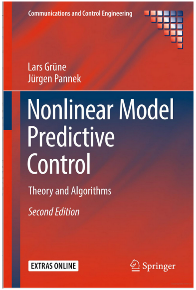








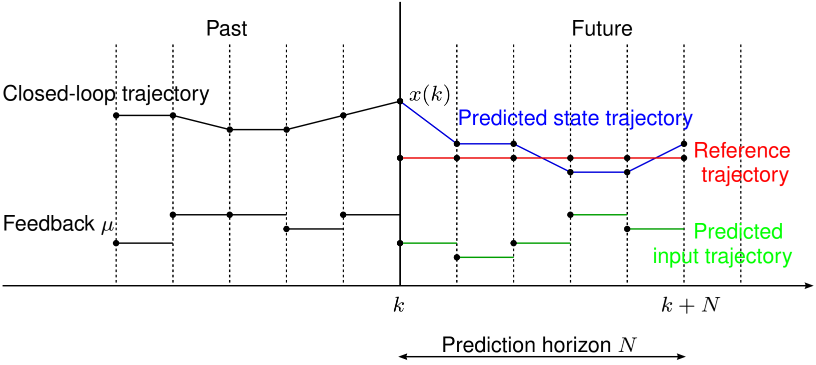




Model Predictive Control

1. MPC Closed-Loop Analysis
   - Performance Estimates
   - Closed Loop Stability Properties
   - Viability & Recursive Feasibility
   - Hard and Soft Constraints
2. Model Predictive Control Schemes
   - Time-Varying Systems & Reference Tracking
   - Linear MPC Versus Nonlinar MPC
   - MPC Without Terminal Costs & Constraints
   - Explicit MPC
   - Economic MPC
3. Implementational Aspects of MPC
   - Warm-Start & Suboptimal MPC
   - Formulation of the Optimization Problem

Model Predictive Control



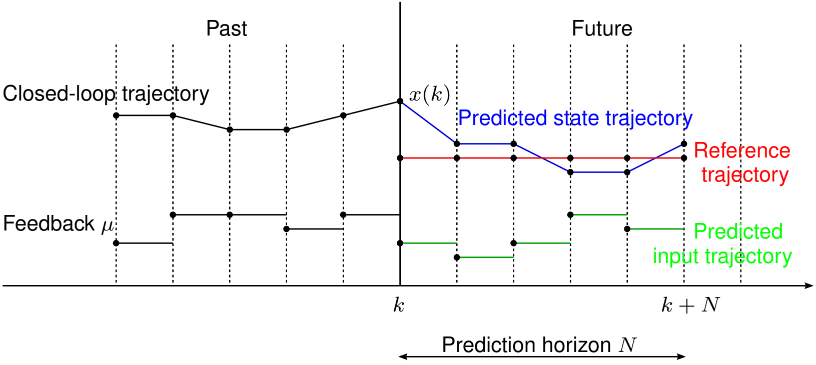
Here, we consider discrete time systems

$$x^+ = f(x,u), \qquad x(0) = x_0 \in \mathbb{R}^n$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ $f(0,0) = 0$.

- State constraints $x \in \mathbb{X} \subset \mathbb{R}^n$
- Input constraints $u \in \mathbb{U}(x) \subset \mathbb{R}^m$

Model Predictive Control



Here, we consider discrete time systems

$$x^+ = f(x,u), \qquad x(0) = x_0 \in \mathbb{R}^n$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ $f(0,0) = 0$.

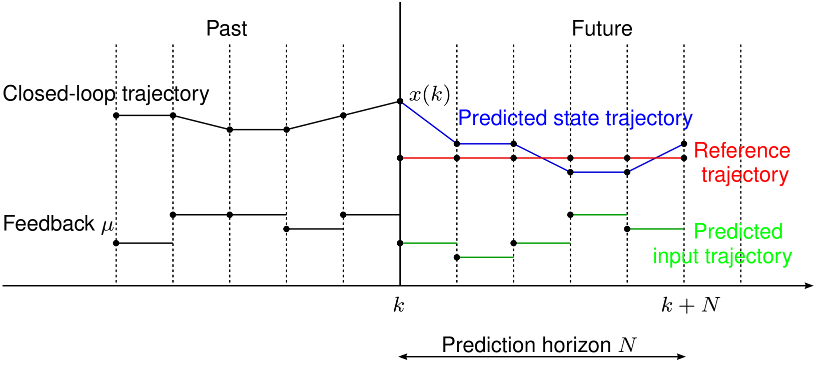
- State constraints $x \in \mathbb{X} \subset \mathbb{R}^n$
- Input constraints $u \in \mathbb{U}(x) \subset \mathbb{R}^m$

- If $\mathbb{U}(x)$ is independent of x we write $\mathbb{U} = \mathbb{U}(x)$
- We combine the state and input constraints through

$$\mathbb{D} = \mathbb{X} \times \mathbb{U}(x)$$

- By assumption $(0,0) \in \mathbb{D}$

Model Predictive Control



MPC is also known as

- *predictive control*
- *receding horizon control*
- *rolling horizon control*

Here, we consider discrete time systems

$$x^+ = f(x,u), \qquad x(0) = x_0 \in \mathbb{R}^n$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ $f(0,0) = 0$.

- State constraints $x \in \mathbb{X} \subset \mathbb{R}^n$
- Input constraints $u \in \mathbb{U}(x) \subset \mathbb{R}^m$

- If $\mathbb{U}(x)$ is independent of x we write $\mathbb{U} = \mathbb{U}(x)$
- We combine the state and input constraints through

$$\mathbb{D} = \mathbb{X} \times \mathbb{U}(x)$$

- By assumption $(0,0) \in \mathbb{D}$

Example: Polyhedral Constraints

For $r_1, r_2 \in \mathbb{N}$ consider

$$\Gamma_{x,1} \in \mathbb{R}^{n\times r_1},\ \Gamma_{x,2} \in \mathbb{R}^{n\times r_2},\ \Gamma_u \in \mathbb{R}^{m\times r_2},$$

$\gamma_1 \in \mathbb{R}^{r_1}$ and $\gamma_2 \in \mathbb{R}^{r_2}$.

Then, state constraints can be described through

$$\mathbb{X} = \{x \in \mathbb{R}^n : \Gamma_{x,1} x \leq \gamma_1\}.$$

For a fixed $x \in \mathbb{X}$, we can define the set (i.e., input constraints)

$$\mathbb{U}(x) = \{u \in \mathbb{R}^m : \Gamma_u u \leq \gamma_2 - \Gamma_{x,2} x\}.$$

The state and input constraints:

$$\mathbb{D} = \left\{ (x,u) \in \mathbb{R}^{n+m} \middle| \begin{bmatrix} \Gamma_{x,1} & 0 \\ \Gamma_{x,2} & \Gamma_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \right\}$$

Example: Polyhedral Constraints

For $r_1, r_2 \in \mathbb{N}$ consider

$$\Gamma_{x,1} \in \mathbb{R}^{n\times r_1},\ \Gamma_{x,2} \in \mathbb{R}^{n\times r_2},\ \Gamma_u \in \mathbb{R}^{m\times r_2},$$
$$\gamma_1 \in \mathbb{R}^{r_1} \text{ and } \gamma_2 \in \mathbb{R}^{r_2}.$$

Then, state constraints can be described through

$$\mathbb{X} = \{x \in \mathbb{R}^n : \Gamma_{x,1} x \leq \gamma_1\}.$$

For a fixed $x \in \mathbb{X}$, we can define the set (i.e., input constraints)

$$\mathbb{U}(x) = \{u \in \mathbb{R}^m : \Gamma_u u \leq \gamma_2 - \Gamma_{x,2} x\}.$$

The state and input constraints:

$$\mathbb{D} = \left\{(x,u) \in \mathbb{R}^{n+m} \,\middle|\, \begin{bmatrix} \Gamma_{x,1} & 0 \\ \Gamma_{x,2} & \Gamma_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \right\}$$

Simple example:

- Consider $\mathbb{X} = [-1,1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.

Example: Polyhedral Constraints

For $r_1, r_2 \in \mathbb{N}$ consider

$$\Gamma_{x,1} \in \mathbb{R}^{n\times r_1},\ \Gamma_{x,2} \in \mathbb{R}^{n\times r_2},\ \Gamma_u \in \mathbb{R}^{m\times r_2},$$
$$\gamma_1 \in \mathbb{R}^{r_1} \text{ and } \gamma_2 \in \mathbb{R}^{r_2}.$$

Then, state constraints can be described through

$$\mathbb{X} = \{x \in \mathbb{R}^n : \Gamma_{x,1} x \le \gamma_1\}.$$

For a fixed $x \in \mathbb{X}$, we can define the set (i.e., input constraints)

$$\mathbb{U}(x) = \{u \in \mathbb{R}^m : \Gamma_u u \le \gamma_2 - \Gamma_{x,2} x\}.$$

The state and input constraints:

$$\mathbb{D} = \left\{ (x,u) \in \mathbb{R}^{n+m} \,\middle|\, \begin{bmatrix} \Gamma_{x,1} & 0 \\ \Gamma_{x,2} & \Gamma_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \le \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \right\}$$

Simple example:

- Consider $\mathbb{X} = [-1,1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Then we can define the matrices and the vectors...

$$\Gamma_{x,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix},\ \gamma_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\Gamma_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \Gamma_{x,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \gamma_2 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}.$$

Example: Polyhedral Constraints

For $r_1, r_2 \in \mathbb{N}$ consider

$$\Gamma_{x,1} \in \mathbb{R}^{n \times r_1},\ \Gamma_{x,2} \in \mathbb{R}^{n \times r_2},\ \Gamma_u \in \mathbb{R}^{m \times r_2},$$
$$\gamma_1 \in \mathbb{R}^{r_1} \text{ and } \gamma_2 \in \mathbb{R}^{r_2}.$$

Then, state constraints can be described through

$$\mathbb{X} = \{x \in \mathbb{R}^n : \Gamma_{x,1} x \leq \gamma_1\}.$$

For a fixed $x \in \mathbb{X}$, we can define the set (i.e., input constraints)

$$\mathbb{U}(x) = \{u \in \mathbb{R}^m : \Gamma_u u \leq \gamma_2 - \Gamma_{x,2} x\}.$$

The state and input constraints:

$$\mathbb{D} = \left\{ (x,u) \in \mathbb{R}^{n+m} \,\middle|\, \begin{bmatrix} \Gamma_{x,1} & 0 \\ \Gamma_{x,2} & \Gamma_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \right\}$$

Simple example:

- Consider $\mathbb{X} = [-1,1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Then we can define the matrices and the vectors...

$$\Gamma_{x,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix},\ \gamma_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\Gamma_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \Gamma_{x,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \gamma_2 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}.$$

- ... and combine them

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

to obtain a representation for $\mathbb{D}$

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^{N} = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

(with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

  (with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \,\middle|\, \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

  (with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$
- Optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

($\rightsquigarrow$ finite dimensional optimization problem if N is finite)

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

(with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$
- Optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

($\rightsquigarrow$ finite dimensional optimization problem if N is finite)

- Even if $V_N : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is not known explicitly, for a given $x_0 \in \mathbb{R}^n$, the function $V_N(\cdot)$ can be evaluated in x_0 by solving the OCP.

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \textstyle\sum_{i=0}^{N-1} \ell(x(i), u(i))$$

(with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$
- Optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

($\rightsquigarrow$ finite dimensional optimization problem if N is finite)

- Even if $V_N : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is not known explicitly, for a given $x_0 \in \mathbb{R}^n$, the function $V_N(\cdot)$ can be evaluated in x_0 by solving the OCP.
- Note that, J_N and V_N are defined as *extended real valued functions* which satisfy $J_N(x_0, u_N(\cdot)) = \infty$ and $V_N(x_0) = \infty$ whenever $\mathcal{U}_{\mathbb{D}}^N = \emptyset$ (i.e., when the OCP is infeasible).
- Here and in the following assume that the minimum in the OCP is attained

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

  (with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)
- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$
- Optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

($\rightsquigarrow$ finite dimensional optimization problem if N is finite)

- Even if $V_N : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is not known explicitly, for a given $x_0 \in \mathbb{R}^n$, the function $V_N(\cdot)$ can be evaluated in x_0 by solving the OCP.
- Note that, J_N and V_N are defined as *extended real valued functions* which satisfy $J_N(x_0, u_N(\cdot)) = \infty$ and $V_N(x_0) = \infty$ whenever $\mathcal{U}_{\mathbb{D}}^N = \emptyset$ (i.e., when the OCP is infeasible).
- Here and in the following assume that the minimum in the OCP is attained
- *Optimal open-loop input trajectory* $u_N^\star(\cdot; x_0) \in \mathcal{U}_{\mathbb{D}}^N$ s.t. $x(N) \in \mathbb{X}_F$ &

$$V_N(x_0) = J_N(x_0, u_N^\star(\cdot; x_0)) + F(x(N))$$

The Basic MPC Formulation

- Prediction horizon: $N \in \mathbb{N} \cup \{\infty\}$
- Set of feasible input trajectories of length N (depending on x_0):

$$\mathcal{U}_{\mathbb{D}}^N = \left\{ u_N(\cdot) : \mathbb{N}_{[0,N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k), u(k)), \\ (x(k), u(k)) & \in & \mathbb{D}, \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- We sometimes write $u_N(\cdot; x_0) = u_N(\cdot)$ to highlight the dependence on the initial condition x_0. For clarity, note that

$$u_N(\cdot) = [u_N(0), u_N(1), u(2), \ldots, u_N(N-1)]$$

- *Cost function*: $J_N : \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N \to \mathbb{R} \cup \{\infty\}$,

$$J_N(x_0, u_N(\cdot)) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$$

(with running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$)

- *Terminal cost* $F : \mathbb{R}^n \to \mathbb{R}$ and *terminal constraints* $\mathbb{X}_F \subset \mathbb{R}^n$
- Optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

($\rightsquigarrow$ finite dimensional optimization problem if N is finite)

- Even if $V_N : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is not known explicitly, for a given $x_0 \in \mathbb{R}^n$, the function $V_N(\cdot)$ can be evaluated in x_0 by solving the OCP.
- Note that, J_N and V_N are defined as *extended real valued functions* which satisfy $J_N(x_0, u_N(\cdot)) = \infty$ and $V_N(x_0) = \infty$ whenever $\mathcal{U}_{\mathbb{D}}^N = \emptyset$ (i.e., when the OCP is infeasible).
- Here and in the following assume that the minimum in the OCP is attained
- *Optimal open-loop input trajectory* $u_N^\star(\cdot; x_0) \in \mathcal{U}_{\mathbb{D}}^N$ s.t. $x(N) \in \mathbb{X}_F$ &

$$V_N(x_0) = J_N(x_0, u_N^\star(\cdot; x_0)) + F(x(N))$$

- $u_N^\star(\cdot; x_0)$ is used to iteratively define a *feedback law* μ_N, i.e.,

$$\mu_N(x_0) = u_N^\star(0; x_0)$$

$$x_{\mu_N}(k+1) = f(x_{\mu_N}(k), \mu_N(x(k)))$$

The Basic MPC Formulation (2)

Input: Measurement of the initial condition $x(0)$; prediction horizon $N \in \mathbb{N} \cup \{\infty\}$; running cost $\ell : \mathbb{R}^{n+m} \to \mathbb{R}$; constraints $\mathbb{D} \subset \mathbb{R}^{n+m}$; terminal cost $F : \mathbb{R}^n \to \mathbb{R}$ and terminal constraints $\mathbb{X}_F \subset \mathbb{R}^n$.

For $k = 0, 1, 2, \ldots$

1. Measure the current state of the system $x^+ = f(x, u)$ and define $x_0 = x(k)$.
2. Solve the optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

   to obtain the open-loop input $u_N^\star(\cdot; x_0)$.

3. Define the feedback law

$$\mu_N(x(k)) = u_N^\star(0; x_0).$$

4. Compute $x(k+1) = f(x(k), \mu_N(x(k)))$, increment k to $k+1$ and go to 1.

The Basic MPC Formulation (3)

Note that:

- *Optimal open-loop input trajectory*: $u_N^\star(\cdot; x_0)$
- *Optimal open-loop solution* for $k = 0, \ldots, N-2$

$$x_N^\star(0) = x_0$$
$$x_N^\star(k+1) = f(x_N^\star(k), u_N^\star(k; x_0))$$

- In many applications, the discrete time system is an approximation of a plant

$$\dot{x}_p = f_p(x_p, u), \qquad x_p(0) \in \mathbb{R}^n$$

⇝ In this setting the MPC feedback law is usually defined as a *sample-and-hold feedback*

The Basic MPC Formulation (3)

Note that:

- *Optimal open-loop input trajectory*: $u_N^\star(\cdot; x_0)$
- *Optimal open-loop solution* for $k = 0, \ldots, N-2$

$$x_N^\star(0) = x_0$$
$$x_N^\star(k+1) = f(x_N^\star(k), u_N^\star(k; x_0))$$

- In many applications, the discrete time system is an approximation of a plant

$$\dot{x}_p = f_p(x_p, u), \qquad x_p(0) \in \mathbb{R}^n$$

$\rightsquigarrow$ In this setting the MPC feedback law is usually defined as a *sample-and-hold feedback*

Input: Measurement of the initial condition $x_p(0)$; $N \in \mathbb{N} \cup \{\infty\}$; $\ell : \mathbb{R}^{n+m} \to \mathbb{R}$; $\mathbb{D} \subset \mathbb{R}^{n+m}$; $F : \mathbb{R}^n \to \mathbb{R}$ and $\mathbb{X}_F \subset \mathbb{R}^n$; $\Delta > 0$.

For $k = 0, 1, 2, \ldots$

1. Measure the current state of the plant $\dot{x}_p = f_p(x_p, u)$ and define $x_0 = x_p(k\Delta)$.
2. Solve the optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot) \in \mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$
$$\text{subject to dyn. \& init. cond. and } x(N) \in \mathbb{X}_F$$

   to obtain the open-loop control law $u_N^\star(\cdot; x_0)$.

3. Define the feedback law

$$\mu_N(x_p(k\Delta)) = u_N^\star(0; x_0).$$

4. Apply the feedback law, i.e., for $t \in [k\Delta, (k+1)\Delta)$ solve

$$\dot{x}_p(t) = f_p(x_p(t), \mu_N(x_p(k\Delta))), \quad x_p(k\Delta) \in \mathbb{R}^n,$$

   increment k to $k+1$ and go to 1.

The Basic MPC Formulation (3)

Note that:

- *Optimal open-loop input trajectory*: $u_N^\star(\cdot;x_0)$
- *Optimal open-loop solution* for $k=0,\ldots,N-2$

$$x_N^\star(0) = x_0$$
$$x_N^\star(k+1) = f(x_N^\star(k), u_N^\star(k;x_0))$$

- In many applications, the discrete time system is an approximation of a plant

$$\dot{x}_p = f_p(x_p,u), \qquad x_p(0)\in\mathbb{R}^n$$

⇝ In this setting the MPC feedback law is usually defined as a *sample-and-hold feedback*

Remark

It is not guaranteed that $x_p(\cdot)$ satisfies the state constraints $x_p(t)\in\mathbb{X}$ for all $t\in\mathbb{R}_{\geq 0}$ since the constraints are only enforced at discrete time steps.

Input: Measurement of the initial condition $x_p(0)$; $N\in\mathbb{N}\cup\{\infty\}$; $\ell:\mathbb{R}^{n+m}\to\mathbb{R}$; $\mathbb{D}\subset\mathbb{R}^{n+m}$; $F:\mathbb{R}^n\to\mathbb{R}$ and $\mathbb{X}_F\subset\mathbb{R}^n$; $\Delta>0$.

For $k=0,1,2,\ldots$

1. Measure the current state of the plant $\dot{x}_p=f_p(x_p,u)$ and define $x_0=x_p(k\Delta)$.
2. Solve the optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_{\mathbb{D}}^N} J_N(x_0,u_N(\cdot)) + F(x(N))$$
$$\text{subject to dyn. \& init. cond. and } x(N)\in\mathbb{X}_F$$

   to obtain the open-loop control law $u_N^\star(\cdot;x_0)$.

3. Define the feedback law

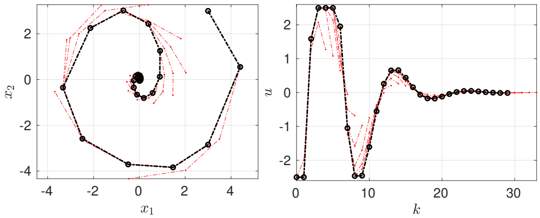
$$\mu_N(x_p(k\Delta)) = u_N^\star(0;x_0).$$

4. Apply the feedback law, i.e., for $t\in[k\Delta,(k+1)\Delta)$ solve

$$\dot{x}_p(t) = f_p(x_p(t),\mu_N(x_p(k\Delta))), \quad x_p(k\Delta)\in\mathbb{R}^n,$$

   increment k to $k+1$ and go to 1.

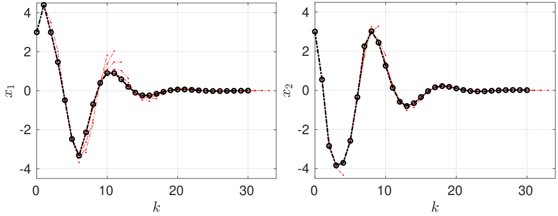
The Basic MPC Formulation (Illustration of properties)



Consider $x^+ = Ax + Bu$ with unstable origin and

$$A = \begin{bmatrix} \frac{6}{5} & \frac{6}{5} \\ -\frac{1}{2} & \frac{6}{5} \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

- Prediction horizon: $N = 5$
- The running cost: $\ell(x, u) = x^T x + 5u^2$
- Constraints: $u \in \mathbb{U} = [-2.5, 2.5]$, $x \in \mathbb{R}^2$ (i.e., $\mathbb{D} = \mathbb{R}^2 \times \mathbb{U}$)
- Terminal cost & constraints: $F(x) = x^T x$, $\mathbb{X}_F = \mathbb{R}^2$.

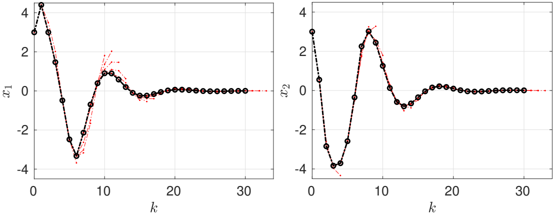


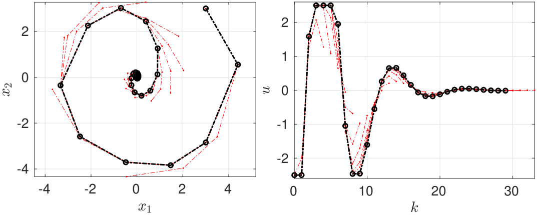
The Basic MPC Formulation (Illustration of properties)

Consider $x^+ = Ax + Bu$ with unstable origin and

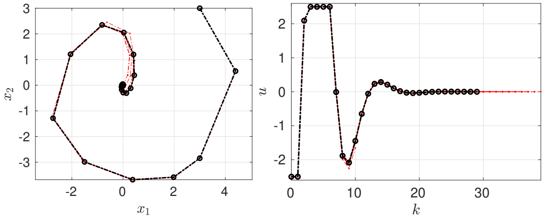
$$A = \begin{bmatrix} \frac{6}{5} & \frac{6}{5} \\ -\frac{1}{2} & \frac{6}{5} \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

- Prediction horizon: $N = 5$
- The running cost: $\ell(x,u) = x^T x + 5u^2$
- Constraints: $u \in \mathbb{U} = [-2.5, 2.5]$, $x \in \mathbb{R}^2$ (i.e., $\mathbb{D} = \mathbb{R}^2 \times \mathbb{U}$)
- Terminal cost & constraints: $F(x) = x^T x$, $\mathbb{X}_F = \mathbb{R}^2$.





- Now, use the terminal constraint $\mathbb{X}_F = \{0\}$ (which makes $F(x)$ superfluous)
- Prediction horizon $N = 11$ (since for $N < 11$ the OCP is not feasible for $x_0 = [3\ 3]^T$)



The Basic MPC Formulation (Illustration of properties, 2)

Consider again $x^+ = Ax + Bu$ with unstable origin and

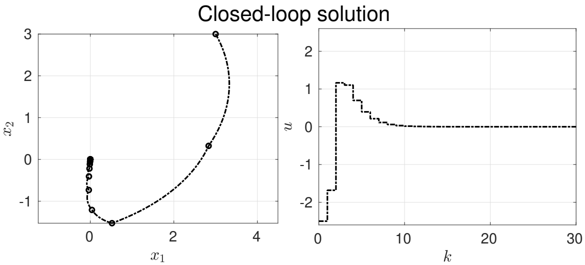
$$A = \begin{bmatrix} \frac{6}{5} & \frac{6}{5} \\ -\frac{1}{2} & \frac{6}{5} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

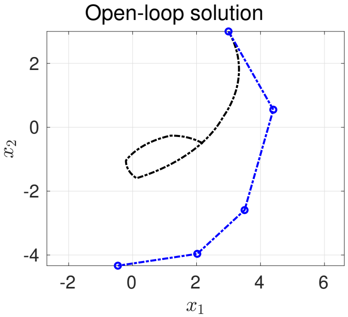
The discrete dynamics define the Euler approximation of

$$\dot{x}_p = A_p x + B_p x = \begin{bmatrix} \frac{1}{5} & \frac{12}{5} \\ -1 & \frac{1}{5} \end{bmatrix} x_p + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u$$

for $\Delta = 0.5$

- Prediction horizon: $N = 5$
- The running cost: $\ell(x,u) = x^T x + 5u^2$
- Constraints: $u \in \mathbb{U} = [-2.5, 2.5]$, $x \in \mathbb{R}^2$
- Terminal cost & constraints: $F(x) = x^T x$, $\mathbb{X}_F = \mathbb{R}^2$.





The Basic MPC Formulation (Illustration of properties, 2)

Consider again $x^+ = Ax + Bu$ with unstable origin and

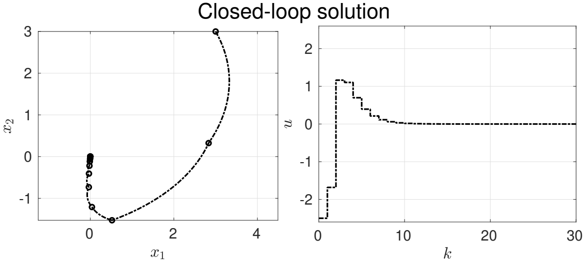
$$A = \begin{bmatrix} \frac{6}{5} & \frac{6}{5} \\ -\frac{1}{2} & \frac{6}{5} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

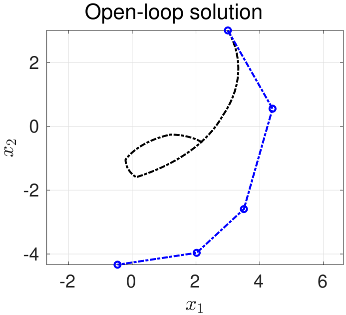
The discrete dynamics define the Euler approximation of

$$\dot{x}_p = A_p x + B_p x = \begin{bmatrix} \frac{1}{5} & \frac{12}{5} \\ -1 & \frac{1}{5} \end{bmatrix} x_p + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u$$

for $\Delta = 0.5$

- Prediction horizon: $N = 5$
- The running cost: $\ell(x,u) = x^T x + 5u^2$
- Constraints: $u \in \mathbb{U} = [-2.5, 2.5]$, $x \in \mathbb{R}^2$
- Terminal cost & constraints: $F(x) = x^T x$, $\mathbb{X}_F = \mathbb{R}^2$.





Remark

Since a rather large Δ is used, the two solutions differ significantly. This highlights an important difference between a feedback law and an open loop control law and provides one explanation why in MPC in general only the first piece of $u_N^\star(\cdot)$ is used to define a feedback law.

Section 1

MPC Closed-Loop Analysis

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.
- Reasonable questions: What is the relation between
  - $V_N(\cdot)$ $(N < \infty)$ and $V_\infty(\cdot)$?
  - the MPC closed-loop performance $J_\infty(x_0, \mu_N(\cdot))$ and $V_\infty(\cdot)$?
- Here, the MPC closed-loop costs are defined as

$$J_\infty(x_0, \mu_N(\cdot)) = \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), u_N^\star(0; x(k))), \quad x(0) = x_0$$

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.
- Reasonable questions: What is the relation between
  - $V_N(\cdot)$ $(N < \infty)$ and $V_\infty(\cdot)$?
  - the MPC closed-loop performance $J_\infty(x_0, \mu_N(\cdot))$ and $V_\infty(\cdot)$?

- Here, the MPC closed-loop costs are defined as

$$J_\infty(x_0, \mu_N(\cdot)) = \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), u_N^\star(0; x(k)), \quad x(0) = x_0$$

- We assume that $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is positive semidefinite
- If $F(x) = 0$ and $\mathbb{X}_F = \mathbb{R}^n$, then it holds that

$$\begin{aligned} V_N(x_0) &= J_N(x_0, u_N^\star(\cdot)) \\ &\leq J_N(x_0, (u_\infty^\star(0), \ldots, u_\infty^\star(N-1))) \\ &\leq J_\infty(x_0, u_\infty^\star(\cdot)) = V_\infty(x_0) \leq J_\infty(x_0, \mu_N(\cdot)) \end{aligned}$$

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.
- Reasonable questions: What is the relation between
  - $V_N(\cdot)$ $(N < \infty)$ and $V_\infty(\cdot)$?
  - the MPC closed-loop performance $J_\infty(x_0, \mu_N(\cdot))$ and $V_\infty(\cdot)$?

- Here, the MPC closed-loop costs are defined as

$$J_\infty(x_0, \mu_N(\cdot)) = \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), u_N^\star(0; x(k)), \quad x(0) = x_0$$

- We assume that $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is positive semidefinite
- If $F(x) = 0$ and $\mathbb{X}_F = \mathbb{R}^n$, then it holds that

$$\begin{aligned} V_N(x_0) &= J_N(x_0, u_N^\star(\cdot)) \\ &\leq J_N(x_0, (u_\infty^\star(0), \ldots, u_\infty^\star(N-1))) \\ &\leq J_\infty(x_0, u_\infty^\star(\cdot)) = V_\infty(x_0) \leq J_\infty(x_0, \mu_N(\cdot)) \end{aligned}$$

- It is in general more interesting to establish bounds

$$J_\infty(x_0, \mu_N(\cdot)) \leq \frac{1}{\alpha_N} V_\infty(x_0) \qquad \forall\, x \in \mathbb{R}^n$$

for an $\alpha_N \in (0, 1]$. $\rightsquigarrow$ level of suboptimality

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.
- Reasonable questions: What is the relation between
  - $V_N(\cdot)$ ($N < \infty$) and $V_\infty(\cdot)$?
  - the MPC closed-loop performance $J_\infty(x_0, \mu_N(\cdot))$ and $V_\infty(\cdot)$?

- Here, the MPC closed-loop costs are defined as

$$J_\infty(x_0, \mu_N(\cdot)) = \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), u_N^\star(0; x(k)), \quad x(0) = x_0$$

- We assume that $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is positive semidefinite
- If $F(x) = 0$ and $\mathbb{X}_F = \mathbb{R}^n$, then it holds that

$$\begin{aligned} V_N(x_0) &= J_N(x_0, u_N^\star(\cdot)) \\ &\leq J_N(x_0, (u_\infty^\star(0), \ldots, u_\infty^\star(N-1))) \\ &\leq J_\infty(x_0, u_\infty^\star(\cdot)) = V_\infty(x_0) \leq J_\infty(x_0, \mu_N(\cdot)) \end{aligned}$$

- It is in general more interesting to establish bounds

$$J_\infty(x_0, \mu_N(\cdot)) \leq \frac{1}{\alpha_N} V_\infty(x_0) \qquad \forall\, x \in \mathbb{R}^n$$

  for an $\alpha_N \in (0, 1]$. $\rightsquigarrow$ level of suboptimality
- For example, if $\alpha_N = \frac{1}{2}$, the MPC closed loop cost is at most twice the infinite horizon optimal control cost.

MPC Closed-Loop Analysis

Advantage:

- MPC can be applied to general nonlinear systems and constraints can be taken into account directly in the controller design.

Disadvantage:

- Since the feedback law is only defined implicitly, the analysis of the closed-loop dynamics is rather difficult.

Performance Analysis:

- Often the OCP solved in every iteration of the MPC algorithm is a compromise between numerical complexity and optimality.
- In many applications, one is interested in solving the OCP for $N = \infty$.
- However the underlying infinite dimensional optimization problem is usually intractable.
- Reasonable questions: What is the relation between
  - $V_N(\cdot)$ ($N < \infty$) and $V_\infty(\cdot)$?
  - the MPC closed-loop performance $J_\infty(x_0, \mu_N(\cdot))$ and $V_\infty(\cdot)$?

- Here, the MPC closed-loop costs are defined as

$$J_\infty(x_0, \mu_N(\cdot)) = \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k), u_N^\star(0; x(k)), \quad x(0) = x_0$$

- We assume that $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is positive semidefinite
- If $F(x) = 0$ and $\mathbb{X}_F = \mathbb{R}^n$, then it holds that

$$\begin{aligned} V_N(x_0) &= J_N(x_0, u_N^\star(\cdot)) \\ &\leq J_N(x_0, (u_\infty^\star(0), \ldots, u_\infty^\star(N-1))) \\ &\leq J_\infty(x_0, u_\infty^\star(\cdot)) = V_\infty(x_0) \leq J_\infty(x_0, \mu_N(\cdot)) \end{aligned}$$

- It is in general more interesting to establish bounds

$$J_\infty(x_0, \mu_N(\cdot)) \leq \frac{1}{\alpha_N} V_\infty(x_0) \qquad \forall\, x \in \mathbb{R}^n$$

  for an $\alpha_N \in (0, 1]$. $\rightsquigarrow$ level of suboptimality
- For example, if $\alpha_N = \frac{1}{2}$, the MPC closed loop cost is at most twice the infinite horizon optimal control cost.
- Under appropriate assumptions, one can expect $\alpha_N \to 1$ for $N \to \infty$.

$\rightsquigarrow$ Out of the scope of this lecture

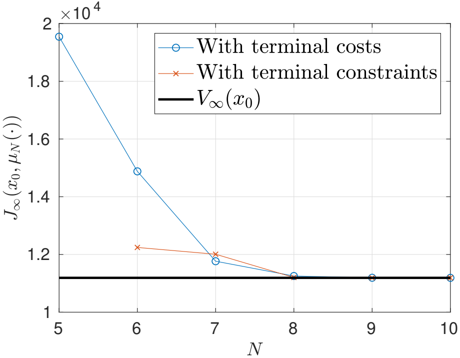
MPC Closed-Loop Analysis (2)

As an example consider:

$$x^+ = Ax + Bu = \begin{bmatrix} 1 & 4 & 0 & 3 & 2 \\ 2 & 4 & 2 & 4 & 2 \\ 3 & 3 & 3 & 0 & 4 \\ 3 & 1 & 3 & 0 & 3 \\ 2 & 3 & 1 & 4 & 4 \end{bmatrix} x + \begin{bmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix} u$$

- $\ell(x,u) = x^T x + u^2;\quad F(x) = x^T x;\quad \mathbb{X}_F = \{0\};$ $\mathbb{U} = [-40, 40]$

Costs corresponding to $x_0 = [1,1,1,1,1]^T$



(To be precise, $V_\infty(x_0)$ is approximated through $V_{1000}(x_0)$)
Note that:

- The plot only shows the costs for a particular initial condition x_0 and thus, it does not provide an estimate with respect to all initial conditions.
- However, for the particular initial condition, small N lead to almost optimal performance.

MPC Closed-Loop Analysis (2)

As an example consider:

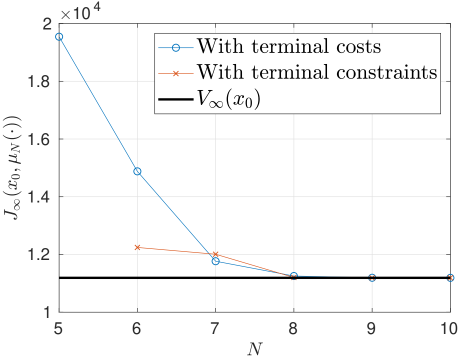
$$x^+ = Ax + Bu = \begin{bmatrix} 1 & 4 & 0 & 3 & 2 \\ 2 & 4 & 2 & 4 & 2 \\ 3 & 3 & 3 & 0 & 4 \\ 3 & 1 & 3 & 0 & 3 \\ 2 & 3 & 1 & 4 & 4 \end{bmatrix} x + \begin{bmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix} u$$

- $\ell(x,u) = x^T x + u^2; \quad F(x) = x^T x; \quad \mathbb{X}_F = \{0\};$ $\mathbb{U} = [-40, 40]$

Remark

The performance estimate discussed here compares the MPC closed loop cost with a particular infinite horizon optimal cost functional. To argue that an MPC controller provides nearly optimal performance (if the parameter α_N is close to 1) while operating a plant is only true with respect to the particular infinite horizon cost functional. Thus, the selection of the running cost needs to be well justified when talking about optimality of a controller.

Costs corresponding to $x_0 = [1,1,1,1,1]^T$



(To be precise, $V_\infty(x_0)$ is approximated through $V_{1000}(x_0)$)
Note that:

- The plot only shows the costs for a particular initial condition x_0 and thus, it does not provide an estimate with respect to all initial conditions.
- However, for the particular initial condition, small N lead to almost optimal performance.

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x, u) = (x - x^e)^T Q(x - x^e) + (u - u^e)^T R(u - u^e)$$

$\rightsquigarrow$ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x,u) = (x - x^e)^T Q(x - x^e) + (u - u^e)^T R(u - u^e)$$

⇝ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

A sufficient condition:

- Stability follows if V_N is a Lyapunov function, i.e.,

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{x^e\}$$

- Even though V_N and μ_N are only known implicitly, conditions on f, $N \in \mathbb{N} \cup \{\infty\}$, ℓ, F and $\mathbb{X}_F$ can be derived, to ensure that V_N is a Lyapunov function

⇝ Often relies on the *principle of optimality* and *dynamic programming*

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x,u) = (x - x^e)^T Q(x - x^e) + (u - u^e)^T R(u - u^e)$$

⇝ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

A sufficient condition:

- Stability follows if V_N is a Lyapunov function, i.e.,

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \ \in \mathbb{X} \backslash \{x^e\}$$

- Even though V_N and μ_N are only known implicitly, conditions on f, $N \in \mathbb{N} \cup \{\infty\}$, ℓ, F and $\mathbb{X}_F$ can be derived, to ensure that V_N is a Lyapunov function

⇝ Often relies on the *principle of optimality* and *dynamic programming*

A sufficient condition using $\mathbb{X}_F = \{x^e\}$:

- If $\mathbb{X}_F = \{x^e\}$ and $\ell(x,u) > \ell(0,0) = 0$ for all $(x,u) \neq (x^e, u^e)$ then V_N is a Lyapunov function

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x,u) = (x - x^e)^T Q (x - x^e) + (u - u^e)^T R (u - u^e)$$

⇝ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

A sufficient condition:

- Stability follows if V_N is a Lyapunov function, i.e.,

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{x^e\}$$

- Even though V_N and μ_N are only known implicitly, conditions on f, $N \in \mathbb{N} \cup \{\infty\}$, ℓ, F and $\mathbb{X}_F$ can be derived, to ensure that V_N is a Lyapunov function

⇝ Often relies on the *principle of optimality* and *dynamic programming*

A sufficient condition using $\mathbb{X}_F = \{x^e\}$:

- If $\mathbb{X}_F = \{x^e\}$ and $\ell(x,u) > \ell(0,0) = 0$ for all $(x,u) \neq (x^e, u^e)$ then V_N is a Lyapunov function
- (W.l.o.g. we assume that $(x^e, u^e) = (0,0)$) Then for all $x_0 \in \mathbb{X}$ it holds that

$$V_N(x_0) = J_N(x_0, u_N^\star(\cdot; x_0)) = \sum_{i=0}^{N-1} \ell(x(i), u_N^\star(i; x_0))$$

$$= \ell(x(0), u_N^\star(0; x_0)) + \sum_{i=1}^{N-1} \ell(x(i), u_N^\star(i; x_0)) + \ell(x(N), 0)$$

$$\geq \ell(x(0), u_N^\star(0; x_0)) + V_N(f(x_0, u_N^\star(0; x_0))).$$

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x,u) = (x - x^e)^T Q(x - x^e) + (u - u^e)^T R(u - u^e)$$

⇝ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

A sufficient condition:

- Stability follows if V_N is a Lyapunov function, i.e.,

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{x^e\}$$

- Even though V_N and μ_N are only known implicitly, conditions on f, $N \in \mathbb{N} \cup \{\infty\}$, ℓ, F and $\mathbb{X}_F$ can be derived, to ensure that V_N is a Lyapunov function

⇝ Often relies on the *principle of optimality* and *dynamic programming*

A sufficient condition using $\mathbb{X}_F = \{x^e\}$:

- If $\mathbb{X}_F = \{x^e\}$ and $\ell(x,u) > \ell(0,0) = 0$ for all $(x,u) \neq (x^e, u^e)$ then V_N is a Lyapunov function
- (W.l.o.g. we assume that $(x^e, u^e) = (0,0)$) Then for all $x_0 \in \mathbb{X}$ it holds that

$$V_N(x_0) = J_N(x_0, u_N^\star(\cdot; x_0)) = \sum_{i=0}^{N-1} \ell(x(i), u_N^\star(i; x_0))$$

$$= \ell(x(0), u_N^\star(0; x_0)) + \sum_{i=1}^{N-1} \ell(x(i), u_N^\star(i; x_0)) + \ell(x(N), 0)$$

$$\geq \ell(x(0), u_N^\star(0; x_0)) + V_N(f(x_0, u_N^\star(0; x_0))).$$

⇝ Since $\ell(x_0, u) > 0$ for $x_0 \neq 0$ it follows that

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{0\}$$

Closed Loop Stability Properties

Consider:

$$x^+ = f(x, \mu_N(x))$$

A standard control application of MPC:

- Stabilization of an equilibrium pair $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$
- Reasonable running costs: ($Q \geq 0$, $R \geq 0$):

$$\ell(x,u) = (x - x^e)^T Q (x - x^e) + (u - u^e)^T R (u - u^e)$$

⇝ How to ensure asymptotic stability of x^e (if $\mu_N(\cdot)$ is not known explicitly)?

A sufficient condition:

- Stability follows if V_N is a Lyapunov function, i.e.,

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{x^e\}$$

- Even though V_N and μ_N are only known implicitly, conditions on f, $N \in \mathbb{N} \cup \{\infty\}$, ℓ, F and $\mathbb{X}_F$ can be derived, to ensure that V_N is a Lyapunov function

⇝ Often relies on the *principle of optimality* and *dynamic programming*

A sufficient condition using $\mathbb{X}_F = \{x^e\}$:

- If $\mathbb{X}_F = \{x^e\}$ and $\ell(x,u) > \ell(0,0) = 0$ for all $(x,u) \neq (x^e, u^e)$ then V_N is a Lyapunov function
- (W.l.o.g. we assume that $(x^e, u^e) = (0,0)$) Then for all $x_0 \in \mathbb{X}$ it holds that

$$V_N(x_0) = J_N(x_0, u_N^\star(\cdot; x_0)) = \sum_{i=0}^{N-1} \ell(x(i), u_N^\star(i; x_0))$$

$$= \ell(x(0), u_N^\star(0; x_0)) + \sum_{i=1}^{N-1} \ell(x(i), u_N^\star(i; x_0)) + \ell(x(N), 0)$$

$$\geq \ell(x(0), u_N^\star(0; x_0)) + V_N(f(x_0, u_N^\star(0; x_0))).$$

⇝ Since $\ell(x_0, u) > 0$ for $x_0 \neq 0$ it follows that

$$V_N(f(x, \mu_N(x))) < V_N(x) \quad \forall x \in \mathbb{X} \backslash \{0\}$$

- However: Here, we have assumed (or need to assume) that the optimization problem is feasible for all initial values $x_0 \in \mathbb{X}$!

Closed Loop Stability Properties (Example)

As an example consider:

$$x^+ = Ax + Bu = \begin{bmatrix} 1 & 4 & 0 & 3 & 2 \\ 2 & 4 & 2 & 4 & 2 \\ 3 & 3 & 3 & 0 & 4 \\ 3 & 1 & 3 & 0 & 3 \\ 2 & 3 & 1 & 4 & 4 \end{bmatrix} x + \begin{bmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix} u$$

- $\ell(x,u) = x^T x + u^2$; $F(x) = x^T x$; $\mathbb{X}_F = \{0\}$; $\mathbb{U} = [-40, 40]$
- $x_0 = [1,1,1,1,1]^T$

Without terminal constraints:

- Feasibility is guaranteed for all $N \in \mathbb{N}$
- V_N is only strictly decreasing for $N \geq 10$

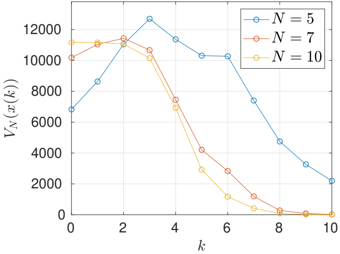
With terminal constraints:

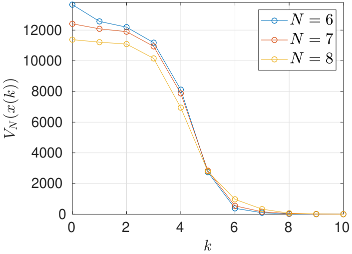
- Feasibility only guaranteed for $N \geq 6$
- V_N is strictly decreasing for all $N \geq 6$ (as expected)

Note that:

- Here, we only look at one initial condition!
- The observations are not necessarily satisfied for all x_0!

Open loop costs $V_N(x(k))$ without terminal constraints and with terminal constraints





Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\rightsquigarrow$ We need to discuss *viability* and *recursive feasibility*.

Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\leadsto$ We need to discuss *viability* and *recursive feasibility*.

Definition (Viability)

Consider $x^+ = f(x,u)$ together with $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U}(x) \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$. The set $\mathbb{X}$ is called viable if

$$\forall\, x \in \mathbb{X} \quad \exists\, u \in \mathbb{U}(x) \ \text{ such that } \ f(x,u) \in \mathbb{X}.$$

A viable set $\mathbb{X}$ is also called a *control invariant set*.

Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\rightsquigarrow$ We need to discuss *viability* and *recursive feasibility*.

Definition (Viability)

Consider $x^+ = f(x, u)$ together with $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U}(x) \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$. The set $\mathbb{X}$ is called viable if

$$\forall\, x \in \mathbb{X} \quad \exists\, u \in \mathbb{U}(x) \ \text{ such that } \ f(x, u) \in \mathbb{X}.$$

A viable set $\mathbb{X}$ is also called a *control invariant set*.

Example

For $a \in \mathbb{R}$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\rightsquigarrow$ We need to discuss *viability* and *recursive feasibility*.

Definition (Viability)

Consider $x^+ = f(x,u)$ together with $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U}(x) \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$. The set $\mathbb{X}$ is called viable if

$$\forall\, x \in \mathbb{X} \quad \exists\, u \in \mathbb{U}(x) \text{ such that } f(x,u) \in \mathbb{X}.$$

A viable set $\mathbb{X}$ is also called a *control invariant set*.

Example

For $a \in \mathbb{R}$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

Example (continued)

Case 1: $|a| \leq 1$

- The origin is asymptotically stable (for $u = 0$)
- For $u = 0$ it holds that $|x^+| \leq |x| \leq 1$
- $\forall\, x \in \mathbb{X}\, \exists\, u \in \mathbb{U}$ (namely $u = 0$) such that $x^+ \in \mathbb{X}$.

Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\rightsquigarrow$ We need to discuss *viability* and *recursive feasibility*.

Definition (Viability)

Consider $x^+ = f(x,u)$ together with $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U}(x) \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$. The set $\mathbb{X}$ is called viable if

$$\forall\, x \in \mathbb{X} \quad \exists\, u \in \mathbb{U}(x) \ \text{ such that } \ f(x,u) \in \mathbb{X}.$$

A viable set $\mathbb{X}$ is also called a *control invariant set*.

Example

For $a \in \mathbb{R}$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1,1]$ and $\mathbb{U} = [-1,1]$

Example (continued)

Case 1: $|a| \leq 1$

- The origin is asymptotically stable (for $u = 0$)
- For $u = 0$ it holds that $|x^+| \leq |x| \leq 1$
- $\forall\, x \in \mathbb{X}\ \exists\, u \in \mathbb{U}$ (namely $u = 0$) such that $x^+ \in \mathbb{X}$.

Case 2: $|a| \in (1,2]$

- Define $u(x) = -\operatorname{sign}(a)x$
- Then, for all $x \in \mathbb{X}$, x^+ satisfies

$$\begin{aligned} |x^+| &= |ax - \operatorname{sign}(a)x| = |a - \operatorname{sign}(a)| \cdot |x| \\ &= ||a| - 1| \cdot |x| \leq |x| \leq 1 \rightsquigarrow \mathbb{X} \text{ is viable} \end{aligned}$$

Viability & Recursive Feasibility

Note that:

- If $\mathbb{X} \neq \mathbb{R}^n$ then the OCP may be infeasible.
- To define implementable feedback laws it is necessary that the OCP is feasible for all $k \in \mathbb{N}$.

$\rightsquigarrow$ We need to discuss *viability* and *recursive feasibility*.

Definition (Viability)

Consider $x^+ = f(x,u)$ together with $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U}(x) \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$. The set $\mathbb{X}$ is called viable if

$$\forall\, x \in \mathbb{X} \quad \exists\, u \in \mathbb{U}(x) \text{ such that } f(x,u) \in \mathbb{X}.$$

A viable set $\mathbb{X}$ is also called a *control invariant set*.

Example

For $a \in \mathbb{R}$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

Example (continued)

Case 1: $|a| \leq 1$

- The origin is asymptotically stable (for $u = 0$)
- For $u = 0$ it holds that $|x^+| \leq |x| \leq 1$
- $\forall\, x \in \mathbb{X} \;\exists\, u \in \mathbb{U}$ (namely $u = 0$) such that $x^+ \in \mathbb{X}$.

Case 2: $|a| \in (1, 2]$

- Define $u(x) = -\operatorname{sign}(a)x$
- Then, for all $x \in \mathbb{X}$, x^+ satisfies

$$\begin{aligned} |x^+| &= |ax - \operatorname{sign}(a)x| = |a - \operatorname{sign}(a)| \cdot |x| \\ &= ||a| - 1| \cdot |x| \leq |x| \leq 1 \rightsquigarrow \mathbb{X} \text{ is viable} \end{aligned}$$

Case 3: $|a| > 2$

- Consider $x = \operatorname{sign}(a)$.
- For $u = 0$, x^+ satisfies $x^+ = a \operatorname{sign}(a) = |a| > 2$.
- The best we can is to select $u = -1$. Thus $x^+ = a \operatorname{sign}(a) - 1 = |a| - 1 > 1$

$\rightsquigarrow$ For $|a| > 2$, the set $\mathbb{X}$ is not viable.

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

$\rightsquigarrow$ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

$\rightsquigarrow$ Is it possible to find a viable subset $\mathbb{X}_\mathrm{v} \subset \mathbb{X}$?

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

⇝ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

⇝ Is it possible to find a viable subset $\mathbb{X}_{\mathrm{v}} \subset \mathbb{X}$?

Example

We continue with the Example and consider

$$x^+ = 3x + u, \quad \mathbb{X} = [-1, 1], \quad \mathbb{U} = [-1, 1]$$

- Since $x = 0 \in \mathbb{X}$ is an equilibrium of the system (and $u = 0 \in \mathbb{U}$), the set $\{0\} \subset \mathbb{X}$ is trivially viable.
- Is it possible to enlarge the viable set and what is its maximal size?
- Assume that $\mathbb{X}_{\mathrm{v}} = [-c_1, c_2]$ for unknown constants $c_1, c_2 \in [0, 1]$.

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

$\rightsquigarrow$ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

$\rightsquigarrow$ Is it possible to find a viable subset $\mathbb{X}_\mathrm{v} \subset \mathbb{X}$?

Example

We continue with the Example and consider

$$x^+ = 3x + u, \quad \mathbb{X} = [-1, 1], \quad \mathbb{U} = [-1, 1]$$

- Since $x = 0 \in \mathbb{X}$ is an equilibrium of the system (and $u = 0 \in \mathbb{U}$), the set $\{0\} \subset \mathbb{X}$ is trivially viable.
- Is it possible to enlarge the viable set and what is its maximal size?
- Assume that $\mathbb{X}_\mathrm{v} = [-c_1, c_2]$ for unknown constants $c_1, c_2 \in [0, 1]$.

Example (continued)

- Then, for $x = c_2$, there needs to exist a $u \in \mathbb{U}$ such that $x^+ \in [-c_1, c_2]$.

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

⇝ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

⇝ Is it possible to find a viable subset $\mathbb{X}_v \subset \mathbb{X}$?

Example

We continue with the Example and consider

$$x^+ = 3x + u, \quad \mathbb{X} = [-1, 1], \quad \mathbb{U} = [-1, 1]$$

- Since $x = 0 \in \mathbb{X}$ is an equilibrium of the system (and $u = 0 \in \mathbb{U}$), the set $\{0\} \subset \mathbb{X}$ is trivially viable.
- Is it possible to enlarge the viable set and what is its maximal size?
- Assume that $\mathbb{X}_v = [-c_1, c_2]$ for unknown constants $c_1, c_2 \in [0, 1]$.

Example (continued)

- Then, for $x = c_2$, there needs to exist a $u \in \mathbb{U}$ such that $x^+ \in [-c_1, c_2]$.
- Since $a > 0$, and $c_2 \geq 0$ by assumption, in the worst case it is only possible to guarantee that $x^+ = c_2$ but $x^+ \notin (-c_1, c_2)$.

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

$\rightsquigarrow$ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

$\rightsquigarrow$ Is it possible to find a viable subset $\mathbb{X}_{\mathrm{v}} \subset \mathbb{X}$?

Example

We continue with the Example and consider

$$x^+ = 3x + u, \quad \mathbb{X} = [-1, 1], \quad \mathbb{U} = [-1, 1]$$

- Since $x = 0 \in \mathbb{X}$ is an equilibrium of the system (and $u = 0 \in \mathbb{U}$), the set $\{0\} \subset \mathbb{X}$ is trivially viable.
- Is it possible to enlarge the viable set and what is its maximal size?
- Assume that $\mathbb{X}_{\mathrm{v}} = [-c_1, c_2]$ for unknown constants $c_1, c_2 \in [0, 1]$.

Example (continued)

- Then, for $x = c_2$, there needs to exist a $u \in \mathbb{U}$ such that $x^+ \in [-c_1, c_2]$.
- Since $a > 0$, and $c_2 \geq 0$ by assumption, in the worst case it is only possible to guarantee that $x^+ = c_2$ but $x^+ \notin (-c_1, c_2)$.
- This leads to the condition

$$c_2 = 3c_2 + u = 3c_2 - 1 \quad \rightsquigarrow \quad c_2 = \tfrac{1}{2}$$

Moreover, the selection of $u = -1$ implies that

$$x^+ = 3x - 1 > c_2 \qquad \forall\, x > c_2 \qquad \text{and}$$
$$x^+ = 3x - 1 < c_2 \qquad \forall\, x < c_2.$$

Viability & Recursive Feasibility (2)

For $|a| > 2$, consider $\quad x^+ = ax + u$

- $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-1, 1]$

⇝ The set $\mathbb{X} = [-1, 1]$ is not viable, i.e., there exist $x_0 \in \mathbb{X}$ such that every corresponding trajectory $x(\cdot; x_0)$ necessarily leaves the domain $\mathbb{X}$.

⇝ Is it possible to find a viable subset $\mathbb{X}_\mathrm{v} \subset \mathbb{X}$?

Example

We continue with the Example and consider

$$x^+ = 3x + u, \quad \mathbb{X} = [-1, 1], \quad \mathbb{U} = [-1, 1]$$

- Since $x = 0 \in \mathbb{X}$ is an equilibrium of the system (and $u = 0 \in \mathbb{U}$), the set $\{0\} \subset \mathbb{X}$ is trivially viable.
- Is it possible to enlarge the viable set and what is its maximal size?
- Assume that $\mathbb{X}_\mathrm{v} = [-c_1, c_2]$ for unknown constants $c_1, c_2 \in [0, 1]$.

Example (continued)

- Then, for $x = c_2$, there needs to exist a $u \in \mathbb{U}$ such that $x^+ \in [-c_1, c_2]$.
- Since $a > 0$, and $c_2 \geq 0$ by assumption, in the worst case it is only possible to guarantee that $x^+ = c_2$ but $x^+ \notin (-c_1, c_2)$.
- This leads to the condition

$$c_2 = 3c_2 + u = 3c_2 - 1 \quad \rightsquigarrow \quad c_2 = \tfrac{1}{2}$$

Moreover, the selection of $u = -1$ implies that

$$x^+ = 3x - 1 > c_2 \qquad \forall\, x > c_2 \qquad \text{and}$$
$$x^+ = 3x - 1 < c_2 \qquad \forall\, x < c_2.$$

- For $x \leq 0$ the same arguments (with $u = 1$) lead to $c_1 = c_2$.

⇝ The maximal viable set contained in $\mathbb{X}$ is given by $\mathbb{X}_\mathrm{v} = [-\frac{1}{2}, \frac{1}{2}]$.

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious
- For more general nonlinear systems subject to nonlinear constraints it might be impossible.
- For linear systems subject to polyhedral constraints the Fourier-Motzkin elimination can be used to compute viable sets. (In Matlab the *MPT-toolbox* can be used to perform these calculations.)

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious
- For more general nonlinear systems subject to nonlinear constraints it might be impossible.
- For linear systems subject to polyhedral constraints the Fourier-Motzkin elimination can be used to compute viable sets. (In Matlab the *MPT-toolbox* can be used to perform these calculations.)

Example

Consider

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

and $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious
- For more general nonlinear systems subject to nonlinear constraints it might be impossible.
- For linear systems subject to polyhedral constraints the Fourier-Motzkin elimination can be used to compute viable sets. (In Matlab the *MPT-toolbox* can be used to perform these calculations.)

Example

Consider

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

and $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.

Example (continued)

Alternatively:

$$\Gamma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \gamma_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \Gamma_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \gamma_u = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_{\mathrm{v}}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious
- For more general nonlinear systems subject to nonlinear constraints it might be impossible.
- For linear systems subject to polyhedral constraints the Fourier-Motzkin elimination can be used to compute viable sets. (In Matlab the *MPT-toolbox* can be used to perform these calculations.)

Example

Consider

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

and $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.

Example (continued)

Alternatively:

$$\Gamma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \gamma_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \Gamma_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \gamma_u = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

Idea: Starting with $\mathbb{X}_0 = \mathbb{X}$ iteratively define

$$\mathbb{X}_{i+1} = \{x \in \mathbb{X}_i : \exists u \in \mathbb{U} \text{ such that } Ax + Bu \in \mathbb{X}_i\}$$

If $\mathbb{X}_{i+1} = \mathbb{X}_i$ is satisfied then $\mathbb{X}_{\mathrm{v}} = \mathbb{X}_i$ is viable.

Viability & Recursive Feasibility (3)

Note that:

- For linear one dimensional linear systems it is relatively simple to check viability of $\mathbb{X}$, or to compute a viable set $\mathbb{X}_\text{v}$.
- For higher dimensional linear systems subject to constraints $\mathbb{X}$ and $\mathbb{U}$ the verification of viability quickly becomes tedious
- For more general nonlinear systems subject to nonlinear constraints it might be impossible.
- For linear systems subject to polyhedral constraints the Fourier-Motzkin elimination can be used to compute viable sets. (In Matlab the *MPT-toolbox* can be used to perform these calculations.)

Example

Consider

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

and $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.

Example (continued)

Alternatively:

$$\Gamma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \gamma_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \Gamma_u = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \gamma_u = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

Idea: Starting with $\mathbb{X}_0 = \mathbb{X}$ iteratively define

$$\mathbb{X}_{i+1} = \{x \in \mathbb{X}_i : \exists u \in \mathbb{U} \text{ such that } Ax + Bu \in \mathbb{X}_i\}$$

If $\mathbb{X}_{i+1} = \mathbb{X}_i$ is satisfied then $\mathbb{X}_\text{v} = \mathbb{X}_i$ is viable. Define

$$\widetilde{\mathbb{X}}_i = \{[x^T, u]^T \in \mathbb{R}^3 : \Delta_i (x^T, u)^T \leq \delta_i\} \quad (1)$$

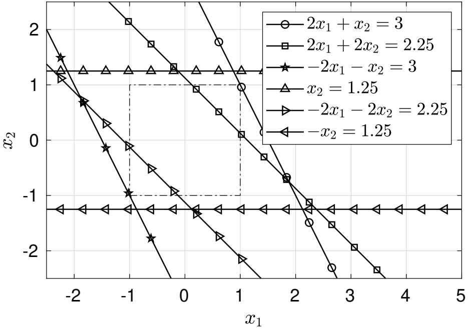
with

$$\Delta_i = \begin{bmatrix} \Gamma_i & 0 \\ \Gamma_i A & \Gamma_i B \\ 0 & \Gamma_u \end{bmatrix} \quad \text{and} \quad \delta_i = \begin{bmatrix} \gamma_i \\ \gamma_i \\ \gamma_u \end{bmatrix}.$$

Then, $\mathbb{X}_{i+1} = P_x(\widetilde{\mathbb{X}}_i)$ is obtained by projecting $\widetilde{\mathbb{X}}_i$ on the (x_1, x_2)-subspace.

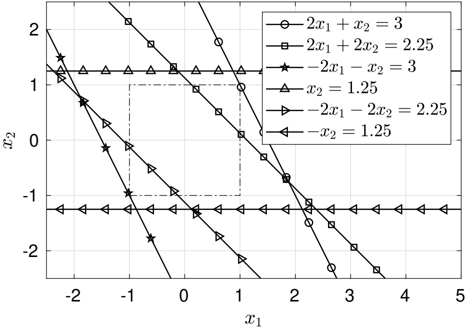
Viability & Recursive Feasibility (4)

The projection $\mathbb{X}_1 = P_x(\widetilde{\mathbb{X}}_0)$ leads to the conditions represented in the figure:

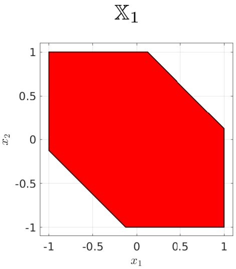


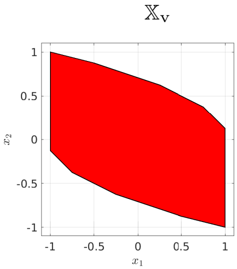
Viability & Recursive Feasibility (4)

The projection $\mathbb{X}_1 = P_x(\widetilde{\mathbb{X}}_0)$ leads to the conditions represented in the figure:



(⇝ Have a look in the lecture notes for details.)





The viable set $\mathbb{X}_{\mathrm{v}}$ is defined through:

$$\begin{bmatrix} -0.24 & -0.97 \\ -0.32 & -0.95 \\ 0.71 & 0.71 \\ 0.32 & 0.95 \\ -0.45 & -0.89 \\ 0.24 & 0.97 \\ 0.45 & 0.89 \\ -0.71 & -0.71 \\ 1.00 & 0.00 \\ -1.00 & 0.00 \end{bmatrix} x \leq \begin{bmatrix} 0.73 \\ 0.67 \\ 0.80 \\ 0.67 \\ 0.67 \\ 0.73 \\ 0.67 \\ 0.80 \\ 1.00 \\ 1.00 \end{bmatrix}$$

Viability & Recursive Feasibility (4)

Definition (Recursive feasibility)

Consider the MPC Algorithm with input constraints $\mathbb{U}$ and a set of initial states $\mathbb{X}_{\mathrm{rf}}^N \subset \mathbb{X}$. The set $\mathbb{X}_{\mathrm{rf}}^N$ is called recursively feasible with respect to the MPC Algorithm and the prediction horizon $N \in \mathbb{N}$ if feasibility of the OCP for $x(0) = x_0 \in \mathbb{X}_{\mathrm{rf}}^N$ implies feasibility of the OCP for all $k \in \mathbb{N}$.

Viability & Recursive Feasibility (4)

Definition (Recursive feasibility)

Consider the MPC Algorithm with input constraints $\mathbb{U}$ and a set of initial states $\mathbb{X}_{\text{rf}}^N \subset \mathbb{X}$. The set $\mathbb{X}_{\text{rf}}^N$ is called recursively feasible with respect to the MPC Algorithm and the prediction horizon $N \in \mathbb{N}$ if feasibility of the OCP for $x(0) = x_0 \in \mathbb{X}_{\text{rf}}^N$ implies feasibility of the OCP for all $k \in \mathbb{N}$.

Example

Consider

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

- $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Running cost $\ell(x, u) = x^T x + 10u^2$
- No terminal cost/constraint
- Initial condition $x_0 = [-1, 1]^T \in \mathbb{X}_{\text{v}}$

Viability & Recursive Feasibility (4)

Definition (Recursive feasibility)

Consider the MPC Algorithm with input constraints $\mathbb{U}$ and a set of initial states $\mathbb{X}_{\mathrm{rf}}^N \subset \mathbb{X}$. The set $\mathbb{X}_{\mathrm{rf}}^N$ is called recursively feasible with respect to the MPC Algorithm and the prediction horizon $N \in \mathbb{N}$ if feasibility of the OCP for $x(0) = x_0 \in \mathbb{X}_{\mathrm{rf}}^N$ implies feasibility of the OCP for all $k \in \mathbb{N}$.

Example

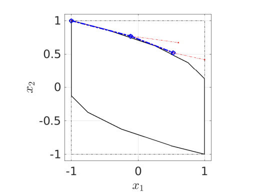
Consider

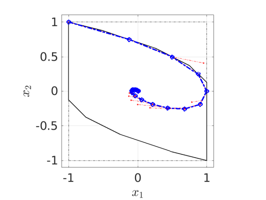
$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

- $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Running cost $\ell(x, u) = x^T x + 10u^2$
- No terminal cost/constraint
- Initial condition $x_0 = [-1, 1]^T \in \mathbb{X}_{\mathrm{v}}$

Example (continued)

- For $N = 3$ we arrive at an infeasible OCP after 2 iterations (left)
- For $N = 4$ (and in fact also for $N \geq 4$), the MPC algorithm does not run into an infeasible optimization problem (right).





Viability & Recursive Feasibility (4)

Definition (Recursive feasibility)

Consider the MPC Algorithm with input constraints $\mathbb{U}$ and a set of initial states $\mathbb{X}_{\text{rf}}^N \subset \mathbb{X}$. The set $\mathbb{X}_{\text{rf}}^N$ is called recursively feasible with respect to the MPC Algorithm and the prediction horizon $N \in \mathbb{N}$ if feasibility of the OCP for $x(0) = x_0 \in \mathbb{X}_{\text{rf}}^N$ implies feasibility of the OCP for all $k \in \mathbb{N}$.

Example

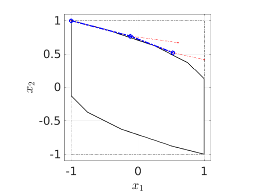
Consider

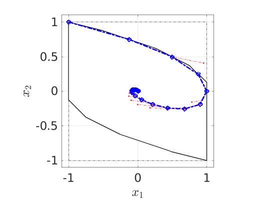
$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

- $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Running cost $\ell(x, u) = x^T x + 10u^2$
- No terminal cost/constraint
- Initial condition $x_0 = [-1, 1]^T \in \mathbb{X}_{\text{v}}$

Example (continued)

- For $N = 3$ we arrive at an infeasible OCP after 2 iterations (left)
- For $N = 4$ (and in fact also for $N \geq 4$), the MPC algorithm does not run into an infeasible optimization problem (right).





Note that:

- If we replace $\mathbb{X}$ by $\mathbb{X}_{\text{v}}$ in the MPC algorithm, then infeasibility is not a problem. (However, this means we need to know $\mathbb{X}_{\text{v}}$.)

Viability & Recursive Feasibility (5)

Note that:

- Recursive feasibility shifts the problem of running into an infeasible optimization problem from viability to recursive feasibility. However, similar to viability, recursive feasibility of a set $\mathbb{X}_{\mathrm{rf}}^N$ is in general nontrivial to establish.

Viability & Recursive Feasibility (5)

Note that:

- Recursive feasibility shifts the problem of running into an infeasible optimization problem from viability to recursive feasibility. However, similar to viability, recursive feasibility of a set $\mathbb{X}_{\mathrm{rf}}^N$ is in general nontrivial to establish.

Lemma

Consider the MPC Algorithm and assume that $\mathbb{U}(x) = \mathbb{U} \subset \mathbb{R}^m$ for all $x \in \mathbb{X}$ and $\mathbb{X}_F = \mathbb{R}^n$. If $\mathbb{X}$ is viable, then $\mathbb{X}_{\mathrm{rf}}^N = \mathbb{X}$ is recursively feasible for all $N \in \mathbb{N}$.

Proof.

- Since $\mathbb{X}$ is viable, for all $x(k) \in \mathbb{X}$ there exist $u(k) \in \mathbb{U}$ such that $x(k+1) \in \mathbb{X}$, $k = 0, \ldots, N-1$.
- If $x(0) = x_0 \in \mathbb{X}$ is satisfied, the OCP is feasible.
- At the next time step, the OCP is initialized through $f(x_0, u^\star(0; x_0)) \in \mathbb{X}$ and the same argument can be applied iteratively.

□

Viability & Recursive Feasibility (5)

Note that:

- Recursive feasibility shifts the problem of running into an infeasible optimization problem from viability to recursive feasibility. However, similar to viability, recursive feasibility of a set $\mathbb{X}_{\mathrm{rf}}^N$ is in general nontrivial to establish.

Lemma

Consider the MPC Algorithm and assume that $\mathbb{U}(x) = \mathbb{U} \subset \mathbb{R}^m$ *for all* $x \in \mathbb{X}$ *and* $\mathbb{X}_F = \mathbb{R}^n$*. If* $\mathbb{X}$ *is viable, then* $\mathbb{X}_{\mathrm{rf}}^N = \mathbb{X}$ *is recursively feasible for all* $N \in \mathbb{N}$*.*

Proof.

- Since $\mathbb{X}$ is viable, for all $x(k) \in \mathbb{X}$ there exist $u(k) \in \mathbb{U}$ such that $x(k+1) \in \mathbb{X}$, $k = 0, \ldots, N-1$.
- If $x(0) = x_0 \in \mathbb{X}$ is satisfied, the OCP is feasible.
- At the next time step, the OCP is initialized through $f(x_0, u^\star(0; x_0)) \in \mathbb{X}$ and the same argument can be applied iteratively.

□

Lemma

Consider the MPC Algorithm and assume that $\mathbb{U}(x) = \mathbb{U} \subset \mathbb{R}^m$ *for all* $x \in \mathbb{X}$*. Additionally assume that* $\mathbb{X}_F \subset \mathbb{X}$ *defines a viable set. If the OCP is feasible for all* $x_0 \in \mathbb{X}_{\mathrm{rf}}^N$*, then* $\mathbb{X}_{\mathrm{rf}}^N = \mathbb{X}$ *is recursively feasible.*

Proof.

- Let the OCP be feasible for all $x_0 \in \mathbb{X}_{\mathrm{rf}}^N$.
- Then there exist $u(k) \in \mathbb{U}$ such that $x(k+1) \in \mathbb{X}$ for all $k = 0, \ldots, N-1$ and $x(N) \in \mathbb{X}_F$.
- Moreover, since $\mathbb{X}_F$ is viable, there exists $u(N) \in \mathbb{U}$ such that $x(N+1) \in \mathbb{X}_F$. In particular $u(1), \ldots, u(N)$ is feasible for the OCP at time $k = 1$ initialized through $x_0 = x(1)$.
- This argument can be applied iteratively showing recursive feasibility.

□

Hard and Soft Constraints

Note that:

- Infeasibility of the OCP can only occur in the presence of state constraints $\mathbb{X} \neq \mathbb{R}^n$.
- In some applications it is justifiable to circumvent infeasible optimization problems by rewriting *hard constraints* as *soft constraints*.

Hard and Soft Constraints

Note that:

- Infeasibility of the OCP can only occur in the presence of state constraints $\mathbb{X} \neq \mathbb{R}^n$.
- In some applications it is justifiable to circumvent infeasible optimization problems by rewriting *hard constraints* as *soft constraints*.

Recall

- The combined state and input constraints: $\mathbb{D} \subset \mathbb{R}^{n+m}$

Define

- Distance to $\mathbb{D}$:

$$d_{\mathbb{D}}(x,u) = \min_{(v,w)\in\mathbb{D}} \sqrt{|x-v|^2 + |u-w|^2}$$

- Distance to the terminal set $\mathbb{X}_F$: $d_F : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$,

$$d_F(x) = \min_{v\in\mathbb{X}_F} |x-v|$$

- Introduce costs: $(\alpha, \alpha_F \in \mathcal{K})$

$$\ell_{\mathsf{s}}(x,u) = \alpha(d_{\mathbb{D}}(x,u)) \quad \text{and} \quad F_{\mathsf{s}}(x) = \alpha_F(d_F(x)).$$

Hard and Soft Constraints

Note that:

- Infeasibility of the OCP can only occur in the presence of state constraints $\mathbb{X} \neq \mathbb{R}^n$.
- In some applications it is justifiable to circumvent infeasible optimization problems by rewriting *hard constraints* as *soft constraints*.

Recall

- The combined state and input constraints: $\mathbb{D} \subset \mathbb{R}^{n+m}$

Define

- Distance to $\mathbb{D}$:

$$d_{\mathbb{D}}(x,u) = \min_{(v,w)\in\mathbb{D}} \sqrt{|x-v|^2 + |u-w|^2}$$

- Distance to the terminal set $\mathbb{X}_F$: $d_F : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$,

$$d_F(x) = \min_{v\in\mathbb{X}_F} |x-v|$$

- Introduce costs: $(\alpha, \alpha_F \in \mathcal{K})$

$$\ell_{\mathsf{s}}(x,u) = \alpha(d_{\mathbb{D}}(x,u)) \quad \text{and} \quad F_{\mathsf{s}}(x) = \alpha_F(d_F(x)).$$

- Redefine feasible input trajectories:

$$\mathcal{U}^N = \left\{ u_N(\cdot) \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k),u(k)) \\ & \forall & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- OCP:

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}^N} J_N(x_0,u_N(\cdot)) + F(x(N)) + \sum_{i=0}^{N-1} \ell_{\mathsf{s}}(x(i),u(i)) + F_{\mathsf{s}}(x(N))$$

$$\text{subject to } x^+ = f(x,u),\ x(0) = x_0,$$

Hard and Soft Constraints

Note that:

- Infeasibility of the OCP can only occur in the presence of state constraints $\mathbb{X} \neq \mathbb{R}^n$.
- In some applications it is justifiable to circumvent infeasible optimization problems by rewriting *hard constraints* as *soft constraints*.

Recall

- The combined state and input constraints: $\mathbb{D} \subset \mathbb{R}^{n+m}$

Define

- Distance to $\mathbb{D}$:

$$d_{\mathbb{D}}(x,u) = \min_{(v,w)\in\mathbb{D}} \sqrt{|x-v|^2 + |u-w|^2}$$

- Distance to the terminal set $\mathbb{X}_F$: $d_F : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$,

$$d_F(x) = \min_{v\in\mathbb{X}_F} |x-v|$$

- Introduce costs: $(\alpha, \alpha_F \in \mathcal{K})$

$$\ell_{\mathsf{s}}(x,u) = \alpha(d_{\mathbb{D}}(x,u)) \quad \text{and} \quad F_{\mathsf{s}}(x) = \alpha_F(d_F(x)).$$

- Redefine feasible input trajectories:

$$\mathcal{U}^N = \left\{ u_N(\cdot) \;\middle|\; \begin{array}{rcl} x(0) & = & x_0, \\ x(k+1) & = & f(x(k),u(k)) \\ \forall & & k \in \mathbb{N}_{[0,N-1]} \end{array} \right\}$$

- OCP:

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$
$$+ \sum_{i=0}^{N-1} \ell_{\mathsf{s}}(x(i),u(i)) + F_{\mathsf{s}}(x(N)))$$
$$\text{subject to } x^+ = f(x,u), \; x(0) = x_0,$$

Note that:

- ⇝ A solution doesn't necessarily satisfy the constraints
- ⇝ The OCP is feasible by construction
- ⇝ $(x,u) \in \mathbb{D}$, $x \in \mathbb{X}_F$ are *hard constraints*, while $\ell_{\mathsf{s}}(x,u)$, $F_{\mathsf{s}}(x)$ in the cost function define *soft constr.*
- ⇝ If $(x,u) \in \mathbb{D}$ & $x \in \mathbb{X}_F$ then $\ell_{\mathsf{s}}(x,u) = 0$ & $F_{\mathsf{s}}(x) = 0$
- ⇝ If $(x,u) \notin \mathbb{D}$ & $x \notin \mathbb{X}_F$ then $\ell_{\mathsf{s}}(x,u) > 0$ & $F_{\mathsf{s}}(x) > 0$ impose additional costs.

Hard and Soft Constraints: Example

Example

Consider

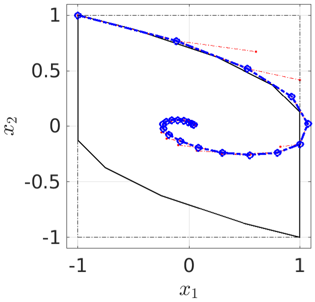
$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

- $\mathbb{X} = [-1,1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Running cost $\ell(x,u) = x^T x + 10u^2$
- No terminal cost/constraint; $N = 3$
- Initial condition $x_0 = [-1,1]^T \in \mathbb{X}_{\mathrm{v}}$

Rewrite hard constraints into soft constraints:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} s \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Penalize $10000 s(i)^T s(i)$ in the cost function



Hard and Soft Constraints: Example

Example

Consider

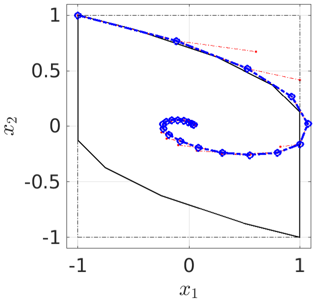
$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = Ax + Bu = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

- $\mathbb{X} = [-1, 1]^2$ and $\mathbb{U} = [-\frac{1}{4}, \frac{1}{4}]$.
- Running cost $\ell(x, u) = x^T x + 10u^2$
- No terminal cost/constraint; $N = 3$
- Initial condition $x_0 = [-1, 1]^T \in \mathbb{X}_{\mathrm{v}}$

Rewrite hard constraints into soft constraints:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} s \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Penalize $10000 s(i)^T s(i)$ in the cost function



Note that, alternatively the constraints

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} s \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

with cost function $10000 s^2$ could have been used, for example.

Section 2

Model Predictive Control Schemes

Model Predictive Control Schemes

Model Predictive Control Schemes: (not a comprehensive list)

- MPC for Time-Varying Systems & Reference Tracking
- Linear MPC
- Nonlinar MPC
- MPC Without Terminal Costs & Constraints (a.k.a. unconstrained MPC)
- Explicit MPC
- Economic MPC
- Robust MPC
- Tube Based MPC
- Stochastic MPC
- Chance constraint MPC
- Distributed MPC
- Multi-step MPC

Model Predictive Control Schemes: Time-Varying Systems & Reference Tracking

Consider: $f : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$$x(k+1) = f(k, x, u), \qquad x(k_0) = x_0 \in \mathbb{R}^n, \; k_0 \in \mathbb{N}_0$$

Model Predictive Control Schemes: Time-Varying Systems & Reference Tracking

Consider: $f : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$$x(k+1) = f(k, x, u), \qquad x(k_0) = x_0 \in \mathbb{R}^n,\ k_0 \in \mathbb{N}_0$$

- Time-varying sets/constraints

$$\mathbb{X}(k) \subset \mathbb{R}^n \quad \text{and} \quad \mathbb{U}(k,x) \subset \mathbb{R}^m \quad \forall k \in \mathbb{N}_0$$
$$\mathbb{D}(k) = \mathbb{X}(k) \times \mathbb{U}(k,x) \subset \mathbb{R}^n \times \mathbb{R}^m \qquad \forall\, k \in \mathbb{N}_0$$

- Set of feasible input trajectories

$$\mathcal{U}_{\mathbb{D}}^N(k) = \left\{ u_N(\cdot;k) : \mathbb{N}_{[k,k+N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{r} x(k) = x_0, \\ x(i+1) = f(i, x(i), u(i)) \\ (x(i), u(i)) \in \mathbb{D}(i), \\ \forall\, i \in \mathbb{N}_{[k,k+N-1]} \end{array} \right\}$$

- Cost function & running cost:

$J_N : \mathbb{N}_0 \times \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N(k) \to \mathbb{R} \cup \{\infty\}, \quad \ell : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$

$$J_N(k, x_0, u_N(\cdot)) = \sum_{i=k}^{k+N-1} \ell(i, x(i), u(i))$$

- Terminal cost & terminal constraints

$$F : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R} \qquad \mathbb{X}_F(k) \subset \mathbb{R}^n, \qquad \forall k \in \mathbb{N}_0$$

- Optimal control problem:

$$V_N(k, x_0) = \min_{u_N(\cdot;k) \in \mathcal{U}_{\mathbb{D}}^N(k)} J_N(k, x_0, u_N(\cdot)) + F(k, x(N))$$
$$\text{s.t. dynamics } \& \ x(N) = \mathbb{X}_F(k)$$

Model Predictive Control Schemes: Time-Varying Systems & Reference Tracking

Consider: $f : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$$x(k+1) = f(k, x, u), \qquad x(k_0) = x_0 \in \mathbb{R}^n,\ k_0 \in \mathbb{N}_0$$

- Time-varying sets/constraints

$$\mathbb{X}(k) \subset \mathbb{R}^n \quad \text{and} \quad \mathbb{U}(k,x) \subset \mathbb{R}^m \quad \forall k \in \mathbb{N}_0$$
$$\mathbb{D}(k) = \mathbb{X}(k) \times \mathbb{U}(k,x) \subset \mathbb{R}^n \times \mathbb{R}^m \qquad \forall\, k \in \mathbb{N}_0$$

- Set of feasible input trajectories

$$\mathcal{U}_{\mathbb{D}}^N(k) = \left\{ u_N(\cdot;k) : \mathbb{N}_{[k,k+N-1]} \to \mathbb{R}^m \,\middle|\, \begin{array}{r} x(k) = x_0, \\ x(i+1) = f(i, x(i), u(i)) \\ (x(i), u(i)) \in \mathbb{D}(i), \\ \forall\, i \in \mathbb{N}_{[k,k+N-1]} \end{array} \right\}$$

- Cost function & running cost:

$J_N : \mathbb{N}_0 \times \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N(k) \to \mathbb{R} \cup \{\infty\}, \quad \ell : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$

$$J_N(k, x_0, u_N(\cdot)) = \sum_{i=k}^{k+N-1} \ell(i, x(i), u(i))$$

- Terminal cost & terminal constraints

$$F : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R} \qquad \mathbb{X}_F(k) \subset \mathbb{R}^n, \qquad \forall k \in \mathbb{N}_0$$

- Optimal control problem:

$$V_N(k, x_0) = \min_{u_N(\cdot;k) \in \mathcal{U}_{\mathbb{D}}^N(k)} J_N(k, x_0, u_N(\cdot)) + F(k, x(N))$$
$$\text{s.t. dynamics } \& \ x(N) = \mathbb{X}_F(k)$$

Input: $k_0 \in \mathbb{N}_0$; $x(k_0)$; $N \in \mathbb{N} \cup \{\infty\}$; $\ell : \mathbb{N}_0 \times \mathbb{R}^{n+m} \to \mathbb{R}$; $\mathbb{D}(k) \subset \mathbb{R}^{n+m}$; $F : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R}$; $\mathbb{X}_F(k) \subset \mathbb{R}^n$.

For $k = k_0, k_0+1, k_0+2, \ldots$

1. Measure the current state and define $x_0 = x(k)$.
2. Solve OCP to obtain open-loop control law $u_N^\star(\cdot; k, x_0)$.
3. Define the feedback law $\mu_N(k, x(k)) = u_N^\star(0; k, x_0)$
4. $x(k+1) = f(k, x(k), \mu_N(k, x(k)))$ increment k to $k+1$ and go to 1.

Model Predictive Control Schemes: Time-Varying Systems & Reference Tracking

Consider: $f : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$$x(k+1) = f(k, x, u), \qquad x(k_0) = x_0 \in \mathbb{R}^n,\ k_0 \in \mathbb{N}_0$$

- Time-varying sets/constraints

$$\mathbb{X}(k) \subset \mathbb{R}^n \quad \text{and} \quad \mathbb{U}(k,x) \subset \mathbb{R}^m \quad \forall k \in \mathbb{N}_0$$
$$\mathbb{D}(k) = \mathbb{X}(k) \times \mathbb{U}(k,x) \subset \mathbb{R}^n \times \mathbb{R}^m \qquad \forall\, k \in \mathbb{N}_0$$

- Set of feasible input trajectories

$$\mathcal{U}_{\mathbb{D}}^N(k) = \left\{ u_N(\cdot; k) : \mathbb{N}_{[k,k+N-1]} \to \mathbb{R}^m \;\middle|\; \begin{array}{r} x(k) = x_0, \\ x(i+1) = f(i, x(i), u(i)) \\ (x(i), u(i)) \in \mathbb{D}(i), \\ \forall\, i \in \mathbb{N}_{[k,k+N-1]} \end{array} \right\}$$

- Cost function & running cost:

$J_N : \mathbb{N}_0 \times \mathbb{R}^n \times \mathcal{U}_{\mathbb{D}}^N(k) \to \mathbb{R} \cup \{\infty\}, \quad \ell : \mathbb{N}_0 \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$

$$J_N(k, x_0, u_N(\cdot)) = \textstyle\sum_{i=k}^{k+N-1} \ell(i, x(i), u(i))$$

- Terminal cost & terminal constraints

$$F : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R} \qquad \mathbb{X}_F(k) \subset \mathbb{R}^n, \qquad \forall k \in \mathbb{N}_0$$

Typical Running costs for reference tracking

$$\ell(k, x, u) = (x - x_{\text{ref}}(k))^T Q (x - x_{\text{ref}}(k)) + (u - u_{\text{ref}}(k))^T R (u - u_{\text{ref}}(k))$$
$$Q \in \mathcal{S}^n_{\geq 0},\ R \in \mathcal{S}^m_{\geq 0}$$

Model Predictive Control Schemes: Linear & Nonlinear MPC

The distinction between linear & nonlinear MPC is not uniform in the literature

Model Predictive Control Schemes: Linear & Nonlinear MPC

The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x, u) \rightsquigarrow$ nonlinear MPC

Model Predictive Control Schemes: Linear & Nonlinear MPC

> The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x, u) \rightsquigarrow$ nonlinear MPC
- Alternatively, based on the OCP:
  - if the OCP is a linear/quadratic program $\rightsquigarrow$ linear MPC
  - if the OCP is a nonlinear program $\rightsquigarrow$ nonlinear MPC

Model Predictive Control Schemes: Linear & Nonlinear MPC

> The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x,u) \rightsquigarrow$ nonlinear MPC
- Alternatively, based on the OCP:
  - if the OCP is a linear/quadratic program $\rightsquigarrow$ linear MPC
  - if the OCP is a nonlinear program $\rightsquigarrow$ nonlinear MPC
- Alternatively, it is common to distinguish between 'simple OCPs' and 'general OCPs':
  - convex optimization $\rightsquigarrow$ linear MPC
  - nonconvex optimization $\rightsquigarrow$ nonlinear MPC

Model Predictive Control Schemes: Linear & Nonlinear MPC

> The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x,u) \rightsquigarrow$ nonlinear MPC
- Alternatively, based on the OCP:
  - if the OCP is a linear/quadratic program $\rightsquigarrow$ linear MPC
  - if the OCP is a nonlinear program $\rightsquigarrow$ nonlinear MPC
- Alternatively, it is common to distinguish between 'simple OCPs' and 'general OCPs':
  - convex optimization $\rightsquigarrow$ linear MPC
  - nonconvex optimization $\rightsquigarrow$ nonlinear MPC

Example (Quadratic Program)

For linear dynamics $x^+ = Ax + Bu$, $Q, P \in \mathcal{S}^n_{\geq 0}$, $R \in \mathcal{S}^m_{\geq 0}$ and polyhedral constraints defined through $\Gamma_x \in \mathbb{R}^{r\times n}$, $\Gamma_u \in \mathbb{R}^{r\times m}$, $\gamma \in \mathbb{R}^r$, $\Gamma_N \in \mathbb{R}^{q\times n}$, $\gamma_N \in \mathbb{R}^q$.
OCP can be written as a QP of the form

$$\min_{\substack{u(i)\in\mathbb{R}^m \\ i\in\mathbb{N}_{[0,N-1]}}} \sum_{i=0}^{N-1} x(i)^T Q x(i) + u(i)^T R u(i) + x(N)^T P x(N)$$

$$\begin{array}{llcll}
\text{subject to} & 0 & = & x(0) - x_0 & \\
& 0 & = & x(i+1) - Ax(i) - Bu(i) & \forall\, i \\
& \gamma & \geq & \Gamma_x x(i) + \Gamma_u u(i) & \forall\, i \\
& \gamma_N & \geq & \Gamma_N x(N) &
\end{array}$$

Model Predictive Control Schemes: Linear & Nonlinear MPC

> The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x,u) \rightsquigarrow$ nonlinear MPC
- Alternatively, based on the OCP:
  - if the OCP is a linear/quadratic program $\rightsquigarrow$ linear MPC
  - if the OCP is a nonlinear program $\rightsquigarrow$ nonlinear MPC
- Alternatively, it is common to distinguish between 'simple OCPs' and 'general OCPs':
  - convex optimization $\rightsquigarrow$ linear MPC
  - nonconvex optimization $\rightsquigarrow$ nonlinear MPC

Example (Convex programs)

Terminal constraints based on a quadratic Lyapunov function, i.e., ($P \in \mathcal{S}^n_{>0}$, $c \in \mathbb{R}_{>0}$)

$$\mathbb{X}_F = \{x \in \mathbb{R}^n : \ x^T P x \leq c\}$$

Convex running cost:

$$\ell(x,u) = (x^T x)^2 + (u^T u)^2$$

Convex optimization problem:

$$\min_{\substack{u(i) \in \mathbb{R}^m \\ i \in \mathbb{N}_{[0,N-1]}}} \sum_{i=0}^{N-1} (x(i)^T x(i))^2 + (u(i)^T u(i))^2$$

$$\begin{array}{lccll} \text{subject to} & 0 & = & x(0) - x_0 & \\ & 0 & = & x(i+1) - Ax(i) - Bu(i) & \forall\, i \\ & \gamma & \geq & \Gamma_x x(i) + \Gamma_u u(i) & \forall\, i \\ & c & \geq & x(N)^T P x(N). & \end{array}$$

Model Predictive Control Schemes: Linear & Nonlinear MPC

> The distinction between linear & nonlinear MPC is not uniform in the literature

For example:

- MPC based on
  - $x^+ = Ax + Bu \rightsquigarrow$ linear MPC
  - $x^+ = f(x,u) \rightsquigarrow$ nonlinear MPC
- Alternatively, based on the OCP:
  - if the OCP is a linear/quadratic program $\rightsquigarrow$ linear MPC
  - if the OCP is a nonlinear program $\rightsquigarrow$ nonlinear MPC
- Alternatively, it is common to distinguish between 'simple OCPs' and 'general OCPs':
  - convex optimization $\rightsquigarrow$ linear MPC
  - nonconvex optimization $\rightsquigarrow$ nonlinear MPC

Example (Nonlinear optimization)

Inverted pendulum on a cart

$$\min_{\substack{u(i)\in\mathbb{R}^m\\ i\in\mathbb{N}_{[0,N-1]}}} \sum_{i=0}^{N-1} x_1^2 + (1-\cos(x_2))^2 + x_3^2 + x_4^2 + u^2$$

$$\begin{array}{llll}
\text{subject to} & 0 = & x(0) - x_0 & \\
 & 0 = & x(i+1) - x(i) - \Delta f(x(i),u(i)) & \forall\, i \\
 & c_u \geq & u(i) & \forall\, i \\
 & c_u \geq & -u(i) & \forall\, i \\
 & c_x \geq & x_1(i) & \forall\, i \\
 & c_x \geq & -x_1(i) & \forall\, i
\end{array}$$

Here $\dot{x} = f(x,u) =$

$$\begin{bmatrix} x_3 \\ x_4 \\ \frac{-\bar{J}\bar{c}x_3 - \bar{J}\sin(x_2)x_4^2 - \bar{\gamma}\cos(x_2)x_4 + g\cos(x_2)\sin(x_2) + \bar{J}u}{\bar{M}\bar{J} - \cos^2(x_2)} \\ \frac{-\bar{M}\bar{\gamma}x_4 + \bar{M}g\sin(x_2) - \bar{c}\cos(x_2)x_3 - \cos(x_2)\sin(x_2)x_4^2 + \cos(x_2)u}{\bar{M}\bar{J} - \cos^2(x_2)} \end{bmatrix}$$

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}_\mathbb{D}^\infty} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_\mathbb{D}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}_{\mathbb{D}}^{\infty}} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_{\mathbb{D}}^{N}} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP

⇝ Compromise in numerical complexity versus optimality

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}_{\mathbb{D}}^\infty} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP

$\rightsquigarrow$ Compromise in numerical complexity versus optimality

- What is a “good” approximation of the infinite horizon OCP?
- How to select N, $\mathbb{X}_F$ and F?
- (Note that this question ignores the problem of how to define good running costs for an infinite horizon optimal control problem in the first place)

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}^\infty_{\mathbb{D}}} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}^N_{\mathbb{D}}} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP

⇝ Compromise in numerical complexity versus optimality

- What is a “good” approximation of the infinite horizon OCP?
- How to select N, $\mathbb{X}_F$ and F?
- (Note that this question ignores the problem of how to define good running costs for an infinite horizon optimal control problem in the first place)
- For linear systems: $\mathbb{X}_F$ and F can be defined based on (sublevel sets of) Lyapunov functions (See Chapter 12). What about nonlinear systems?

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}_{\mathbb{D}}^\infty} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_{\mathbb{D}}^N} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP
- ⇝ Compromise in numerical complexity versus optimality
- What is a "good" approximation of the infinite horizon OCP?
- How to select N, $\mathbb{X}_F$ and F?
- (Note that this question ignores the problem of how to define good running costs for an infinite horizon optimal control problem in the first place)
- For linear systems: $\mathbb{X}_F$ and F can be defined based on (sublevel sets of) Lyapunov functions (See Chapter 12). What about nonlinear systems?
- Are terminal costs/constraints necessary?

Model Predictive Control Schemes: MPC Without Terminal Costs & Constraints

Infinite horizon optimal control problem

$$V_\infty(x_0) = \min_{u_\infty(\cdot)\in\mathcal{U}_{\mathbb{D}}^{\infty}} J_\infty(x_0, u(\cdot))$$

subject to dyn. & init. cond.

Finite horizon optimal control problem

$$V_N(x_0) = \min_{u_N(\cdot)\in\mathcal{U}_{\mathbb{D}}^{N}} J_N(x_0, u_N(\cdot)) + F(x(N))$$

subject to dyn. & init. cond. and $x(N) \in \mathbb{X}_F$

- Often, a finite horizon OCP is used to approximate a corresponding infinite horizon OCP

⇝ Compromise in numerical complexity versus optimality

- What is a “good” approximation of the infinite horizon OCP?
- How to select N, $\mathbb{X}_F$ and F?
- (Note that this question ignores the problem of how to define good running costs for an infinite horizon optimal control problem in the first place)
- For linear systems: $\mathbb{X}_F$ and F can be defined based on (sublevel sets of) Lyapunov functions (See Chapter 12). What about nonlinear systems?
- Are terminal costs/constraints necessary?
  - ‘For sufficiently large prediction horizons $N \in \mathbb{N}$, the MPC closed loop without terminal costs/constraints approximates the corresponding infinite horizon solution arbitrarily well.’
  - Found in publications on ‘MPC Without Terminal Costs & Constraints’ or ‘Unconstrained MPC’

Model Predictive Control Schemes: Explicit MPC

For $k = 0, 1, 2, \ldots$

1. Measure the current state of the system $x^+ = f(x, u)$ and define $x_0 = x(k)$.
2. Solve OCP $\rightsquigarrow$ open-loop input $u_N^\star(\cdot; x_0)$
3. Define $\mu_N(x(k)) = u_N^\star(0; x_0)$
4. Compute $x(k+1) = f(x(k), \mu_N(x(k)))$, increment k to $k+1$ and go to 1.

Note that:

- At every time step an optimization problem needs to be solved
- The optimal value function and the optimal feedback law is only known implicitly

However:

- In some cases it is possible to compute an *explicit* solution of the OCP as a function of x_0.

Model Predictive Control Schemes: Explicit MPC

For $k = 0, 1, 2, \ldots$

1. Measure the current state of the system $x^+ = f(x, u)$ and define $x_0 = x(k)$.
2. Solve OCP $\rightsquigarrow$ open-loop input $u_N^\star(\cdot; x_0)$
3. Define $\mu_N(x(k)) = u_N^\star(0; x_0)$
4. Compute $x(k+1) = f(x(k), \mu_N(x(k)))$, increment k to $k+1$ and go to 1.

Note that:

- At every time step an optimization problem needs to be solved
- The optimal value function and the optimal feedback law is only known implicitly

However:

- In some cases it is possible to compute an *explicit* solution of the OCP as a function of x_0.

Explicit MPC shifts the problem of solving an optimization problem online for all k to a *multiparametric program* which only needs to be solved once and can be solved offline.

Model Predictive Control Schemes: Explicit MPC

For $k = 0, 1, 2, \ldots$

1. Measure the current state of the system $x^+ = f(x,u)$ and define $x_0 = x(k)$.
2. Solve OCP $\rightsquigarrow$ open-loop input $u_N^\star(\cdot; x_0)$
3. Define $\mu_N(x(k)) = u_N^\star(0; x_0)$
4. Compute $x(k+1) = f(x(k), \mu_N(x(k)))$, increment k to $k+1$ and go to 1.

Note that:

- At every time step an optimization problem needs to be solved
- The optimal value function and the optimal feedback law is only known implicitly

However:

- In some cases it is possible to compute an *explicit* solution of the OCP as a function of x_0.

Explicit MPC shifts the problem of solving an optimization problem online for all k to a *multiparametric program* which only needs to be solved once and can be solved offline.

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x,u) = x^2 + u^2$ and $N = 2$:

$$\min_{u(0),u(1)} x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2$$

$$\begin{aligned} \text{subject to} \quad 0 &= x(0) - x_0 \\ 0 &= x(1) - x(0) - 0.5u(0) \\ 0 &= x(2) - x(1) - 0.5u(1) \\ u(0), u(1) &\in [-1, 1]. \end{aligned}$$

- Equivalently

$$\min_{u(0),u(1)} 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

$$\text{subject to} \quad u(0), u(1) \in [-1, 1].$$

Model Predictive Control Schemes: Explicit MPC (2)

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x,u) = x^2 + u^2$ and $N = 2$:

$$
\begin{array}{rrcl}
\min\limits_{u(0),u(1)} & x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2 & & \\
\text{subject to} & 0 & = & x(0) - x_0 \\
 & 0 & = & x(1) - x(0) - 0.5u(0) \\
 & 0 & = & x(2) - x(1) - 0.5u(1) \\
 & u(0), u(1) & \in & [-1, 1].
\end{array}
$$

- Equivalently

$$
\begin{array}{rl}
\min\limits_{u(0),u(1)} & 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2 \\
\text{subject to} & u(0), u(1) \quad \in \quad [-1, 1].
\end{array}
$$

- The gradient of the objective function

$$J_2(u(\cdot); x_0) = 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

is given by

$$\nabla_u J_2(u(\cdot); x_0) = \begin{bmatrix} 2.5u(0) + x_0 \\ 2u(1) \end{bmatrix} \stackrel{!}{=} 0$$

Model Predictive Control Schemes: Explicit MPC (2)

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x,u) = x^2 + u^2$ and $N = 2$:

$$\min_{u(0),u(1)} x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2$$

$$\begin{array}{lrcl} \text{subject to} & 0 & = & x(0) - x_0 \\ & 0 & = & x(1) - x(0) - 0.5u(0) \\ & 0 & = & x(2) - x(1) - 0.5u(1) \\ & u(0), u(1) & \in & [-1, 1]. \end{array}$$

- Equivalently

$$\min_{u(0),u(1)} 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

$$\text{subject to} \quad u(0), u(1) \quad \in \quad [-1, 1].$$

- The gradient of the objective function

$$J_2(u(\cdot); x_0) = 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

is given by

$$\nabla_u J_2(u(\cdot); x_0) = \begin{bmatrix} 2.5u(0) + x_0 \\ 2u(1) \end{bmatrix} \overset{!}{=} 0$$

$\rightsquigarrow$ $u(0) = -0.4x_0$, $u(1) = 0$ if we ignore the constraints

Model Predictive Control Schemes: Explicit MPC (2)

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x,u) = x^2 + u^2$ and $N = 2$:

$$\min_{u(0),u(1)} x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2$$

$$\begin{array}{lrcl} \text{subject to} & 0 & = & x(0) - x_0 \\ & 0 & = & x(1) - x(0) - 0.5u(0) \\ & 0 & = & x(2) - x(1) - 0.5u(1) \\ & u(0), u(1) & \in & [-1, 1]. \end{array}$$

- Equivalently

$$\min_{u(0),u(1)} 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

$$\text{subject to} \quad u(0), u(1) \quad \in \quad [-1, 1].$$

- The gradient of the objective function

$$J_2(u(\cdot); x_0) = 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

is given by

$$\nabla_u J_2(u(\cdot); x_0) = \begin{bmatrix} 2.5u(0) + x_0 \\ 2u(1) \end{bmatrix} \overset{!}{=} 0$$

$\rightsquigarrow$ $u(0) = -0.4x_0$, $u(1) = 0$ if we ignore the constraints

- $x_0 \in [-2.5, 2.5]$ implies $u(0) \in [-1, 1]$

$$\rightsquigarrow \quad \mu(x_0) = -0.4x_0$$

Model Predictive Control Schemes: Explicit MPC (2)

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x, u) = x^2 + u^2$ and $N = 2$:

$$\min_{u(0),u(1)} x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2$$

$$\begin{array}{lrcl} \text{subject to} & 0 & = & x(0) - x_0 \\ & 0 & = & x(1) - x(0) - 0.5u(0) \\ & 0 & = & x(2) - x(1) - 0.5u(1) \\ & u(0), u(1) & \in & [-1, 1]. \end{array}$$

- Equivalently

$$\min_{u(0),u(1)} 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

$$\text{subject to} \quad u(0), u(1) \in [-1, 1].$$

- The gradient of the objective function

$$J_2(u(\cdot); x_0) = 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

is given by

$$\nabla_u J_2(u(\cdot); x_0) = \begin{bmatrix} 2.5u(0) + x_0 \\ 2u(1) \end{bmatrix} \stackrel{!}{=} 0$$

$\rightsquigarrow$ $u(0) = -0.4x_0$, $u(1) = 0$ if we ignore the constraints

- $x_0 \in [-2.5, 2.5]$ implies $u(0) \in [-1, 1]$

$$\rightsquigarrow \quad \mu(x_0) = -0.4x_0$$

- Optimal feedback law: (cont., piecewise affine)

$$\mu_2(x_0) = \begin{cases} 0x_0 + 1 & \text{if } x_0 \leq -2.5 \\ -0.4x_0 + 0 & \text{if } x_0 \in [-2.5, 2.5] \\ 0x_0 - 1 & \text{if } x_0 \geq 2.5 \end{cases}$$

Model Predictive Control Schemes: Explicit MPC (2)

Example

Consider $x^+ = x + 0.5u$ with $u \in [-1, 1]$.

- The origin is stable but not asymptotically stable.
- Every state $x_0 \in \mathbb{R}$ can be driven to the origin in finite time.
- OCP with $\ell(x,u) = x^2 + u^2$ and $N = 2$:

$$\min_{u(0),u(1)} x(0)^2 + x(1)^2 + u(0)^2 + u(1)^2$$

$$\begin{array}{lrcl} \text{subject to} & 0 & = & x(0) - x_0 \\ & 0 & = & x(1) - x(0) - 0.5u(0) \\ & 0 & = & x(2) - x(1) - 0.5u(1) \\ & u(0), u(1) & \in & [-1, 1]. \end{array}$$

- Equivalently

$$\min_{u(0),u(1)} 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

$$\text{subject to} \quad u(0), u(1) \in [-1, 1].$$

- The gradient of the objective function

$$J_2(u(\cdot); x_0) = 2x_0^2 + x_0 u(0) + 1.25u(0)^2 + u(1)^2$$

is given by

$$\nabla_u J_2(u(\cdot); x_0) = \begin{bmatrix} 2.5u(0) + x_0 \\ 2u(1) \end{bmatrix} \stackrel{!}{=} 0$$

$\rightsquigarrow$ $u(0) = -0.4x_0$, $u(1) = 0$ if we ignore the constraints

- $x_0 \in [-2.5, 2.5]$ implies $u(0) \in [-1, 1]$

$$\rightsquigarrow \quad \mu(x_0) = -0.4x_0$$

- Optimal feedback law: (cont., piecewise affine)

$$\mu_2(x_0) = \begin{cases} 0x_0 + 1 & \text{if } x_0 \leq -2.5 \\ -0.4x_0 + 0 & \text{if } x_0 \in [-2.5, 2.5] \\ 0x_0 - 1 & \text{if } x_0 \geq 2.5 \end{cases}$$

- The optimal value function: (cont. differentiable)

$$V_2(x_0) = \begin{cases} 2x_0^2 + x_0 + 1.25 & \text{if} \quad x_0 \leq -2.5 \\ 1.8x_0^2 & \text{if} \quad x_0 \in [-2.5, 2.5] \\ 2x_0^2 - x_0 + 1.25 & \text{if} \quad x_0 \geq 2.5 \end{cases}$$

Model Predictive Control Schemes: Explicit MPC (3)

Example (More general setting)

Consider the linear system $x^+ = Ax + Bu$ defined through

$$A = \begin{bmatrix} 1 & 1 \\ -\frac{1}{4} & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

- Constraints: $x \in [-5, 5]^2$, $u \in [-1, 1]$; Horizon: $N = 5$.
- Running cost, terminal cost: $\ell(x, u) = x^T Q x + u^T R u$, $F(x) = x^T P x$,

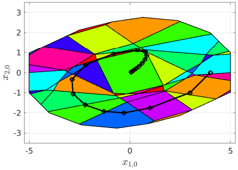
$$Q = P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad R = 1,$$

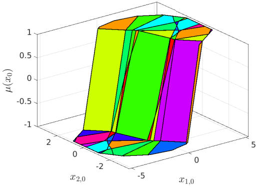
⇝ OCP is a quadratic program

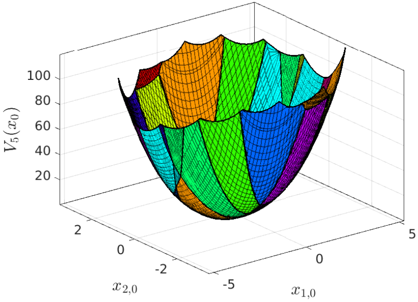
⇝ Feasible region is convex (partition in 53 polyhedral sets)

⇝ $\mu_5(x_0)$ is continuous and piecewise affine

⇝ $V_5(x_0)$ is continuously differentiable







Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

$\rightsquigarrow$ The optimal behavior in terms of a reference point/trajectory needs to be known.

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

⇝ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase “maximize the revenue of a plant” in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

$\rightsquigarrow$ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase “maximize the revenue of a plant” in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

$\rightsquigarrow$ *Economic MPC* investigates the performance of MPC with non-classical positive semidefinite running costs

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

$\rightsquigarrow$ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase "maximize the revenue of a plant" in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

$\rightsquigarrow$ *Economic MPC* investigates the performance of MPC with non-classical positive semidefinite running costs

Note that:

- The MPC Algorithm can be applied independently of the definition of the running cost ℓ, as long as an optimal solution of the OCP can be obtained in every iteration. However, the methods to analyze control schemes with non positive semidefinite running costs are significantly different.

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

⇝ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase "maximize the revenue of a plant" in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

⇝ *Economic MPC* investigates the performance of MPC with non-classical positive semidefinite running costs

Note that:

- The MPC Algorithm can be applied independently of the definition of the running cost ℓ, as long as an optimal solution of the OCP can be obtained in every iteration. However, the methods to analyze control schemes with non positive semidefinite running costs are significantly different.

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\begin{array}{llll} \text{subject to} & 0 = & x(0) - x_0 & \\ & 0 = & x(i+1) - 2x(i) - u(i) & \forall i \end{array}$$

(Maximize revenue x and minimize operational cost u^2.)

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

⇝ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase "maximize the revenue of a plant" in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

⇝ *Economic MPC* investigates the performance of MPC with non-classical positive semidefinite running costs

Note that:

- The MPC Algorithm can be applied independently of the definition of the running cost ℓ, as long as an optimal solution of the OCP can be obtained in every iteration. However, the methods to analyze control schemes with non positive semidefinite running costs are significantly different.

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\begin{array}{llll} \text{subject to} & 0 = & x(0) - x_0 & \\ & 0 = & x(i+1) - 2x(i) - u(i) & \forall i \end{array}$$

(Maximize revenue x and minimize operational cost u^2.)

Note that:

- The OCP is only well defined for $N < \infty$.
- How to compare solution to the infinite horizon OCP?
- However, not the case here since the running cost is not bounded from below.

Model Predictive Control Schemes: Economic MPC

Note that:

- So far we have tacitly assumed that the running cost ℓ is a positive semidefinite function penalizing the distance to a reference trajectory $(x_{\text{ref}}(t), u_{\text{ref}}(t))$.

$\rightsquigarrow$ The optimal behavior in terms of a reference point/trajectory needs to be known.

Questions:

- How to phrase "maximize the revenue of a plant" in this setting?
- Is the closed loop solution converging to an optimal reference point or optimal reference trajectory, even if the optimal point or the optimal trajectory is not known in advance?

$\rightsquigarrow$ *Economic MPC* investigates the performance of MPC with non-classical positive semidefinite running costs

Note that:

- The MPC Algorithm can be applied independently of the definition of the running cost ℓ, as long as an optimal solution of the OCP can be obtained in every iteration. However, the methods to analyze control schemes with non positive semidefinite running costs are significantly different.

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\begin{array}{lrcll} \text{subject to} & 0 & = & x(0) - x_0 & \\ & 0 & = & x(i+1) - 2x(i) - u(i) & \forall i \end{array}$$

(Maximize revenue x and minimize operational cost u^2.)

Note that:

- The OCP is only well defined for $N < \infty$.
- How to compare solution to the infinite horizon OCP?

$\rightsquigarrow$ Average cost: $\bar{V}_\infty(x_0) = \limsup_{K\to\infty} \frac{1}{K} V_K(x_0)$

Even if $V_\infty(x_0) = \infty$, $\bar{V}_\infty(x_0) \in \mathbb{R}$ may hold.

- However, not the case here since the running cost is not bounded from below.

Model Predictive Control Schemes: Economic MPC

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\text{subject to} \quad \begin{aligned} 0 &= x(0) - x_0 \\ 0 &= x(i+1) - 2x(i) - u(i) \qquad \forall i \end{aligned}$$

(Maximize revenue x and minimize operational cost u^2.)

Note that:

- The OCP is only well defined for $N < \infty$.
- How to compare solution to the infinite horizon OCP?

⇝ Average cost: $\bar{V}_\infty(x_0) = \limsup_{K\to\infty} \frac{1}{K} V_K(x_0)$

Even if $V_\infty(x_0) = \infty$, $\bar{V}_\infty(x_0) \in \mathbb{R}$ may hold.

- However, not the case here since the running cost is not bounded from below.

- Consider additional constraints

$$x \in \mathbb{X} = [-2, 2] \qquad \text{and} \qquad u \in \mathbb{U} = [-3, 3]$$

- The optimal average cost satisfies

$$\bar{V}_\infty(x_0) \leq \limsup_{K\to\infty} \tfrac{1}{K} \textstyle\sum_{k=0}^{K-1} 3^2 + 2 = 11$$

$$\bar{V}_\infty(x_0) \geq \limsup_{K\to\infty} \tfrac{1}{K} \textstyle\sum_{k=0}^{K-1} 0^2 - 2 = -2.$$

Model Predictive Control Schemes: Economic MPC

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\begin{array}{llll} \text{subject to} & 0 = & x(0) - x_0 & \\ & 0 = & x(i+1) - 2x(i) - u(i) & \forall i \end{array}$$

(Maximize revenue x and minimize operational cost u^2.)

Note that:

- The OCP is only well defined for $N < \infty$.
- How to compare solution to the infinite horizon OCP?

⇝ Average cost: $\bar{V}_\infty(x_0) = \limsup_{K\to\infty} \frac{1}{K} V_K(x_0)$

Even if $V_\infty(x_0) = \infty$, $\bar{V}_\infty(x_0) \in \mathbb{R}$ may hold.

- However, not the case here since the running cost is not bounded from below.

- Consider additional constraints

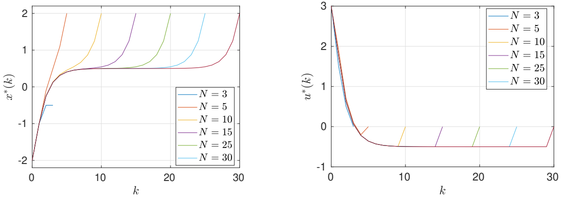
$$x \in \mathbb{X} = [-2, 2] \qquad \text{and} \qquad u \in \mathbb{U} = [-3, 3]$$

- The optimal average cost satisfies

$$\bar{V}_\infty(x_0) \le \limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} 3^2 + 2 = 11$$

$$\bar{V}_\infty(x_0) \ge \limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} 0^2 - 2 = -2.$$

Open-loop solutions for different N



Model Predictive Control Schemes: Economic MPC

Example: Consider

$$x(k+1) = 2x(k) + u(k), \qquad \ell(x,u) = u^2 - x$$

OCP:

$$\min_{u(\cdot)} \sum_{i=0}^{N-1} \left(u(i)^2 - x(i)\right)$$

$$\begin{array}{llll} \text{subject to} & 0 = x(0) - x_0 & & \\ & 0 = x(i+1) - 2x(i) - u(i) & & \forall i \end{array}$$

(Maximize revenue x and minimize operational cost u^2.)

Note that:

- The OCP is only well defined for $N < \infty$.
- How to compare solution to the infinite horizon OCP?

⇝ Average cost: $\bar{V}_\infty(x_0) = \limsup_{K\to\infty} \frac{1}{K} V_K(x_0)$

Even if $V_\infty(x_0) = \infty$, $\bar{V}_\infty(x_0) \in \mathbb{R}$ may hold.

- However, not the case here since the running cost is not bounded from below.

- Consider additional constraints

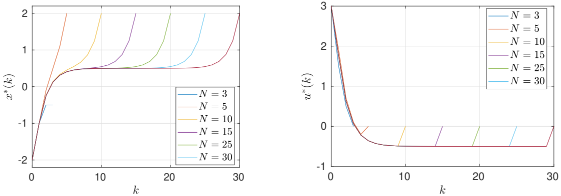
$$x \in \mathbb{X} = [-2, 2] \quad \text{and} \quad u \in \mathbb{U} = [-3, 3]$$

- The optimal average cost satisfies

$$\bar{V}_\infty(x_0) \leq \limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} 3^2 + 2 = 11$$

$$\bar{V}_\infty(x_0) \geq \limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} 0^2 - 2 = -2.$$

Open-loop solutions for different N

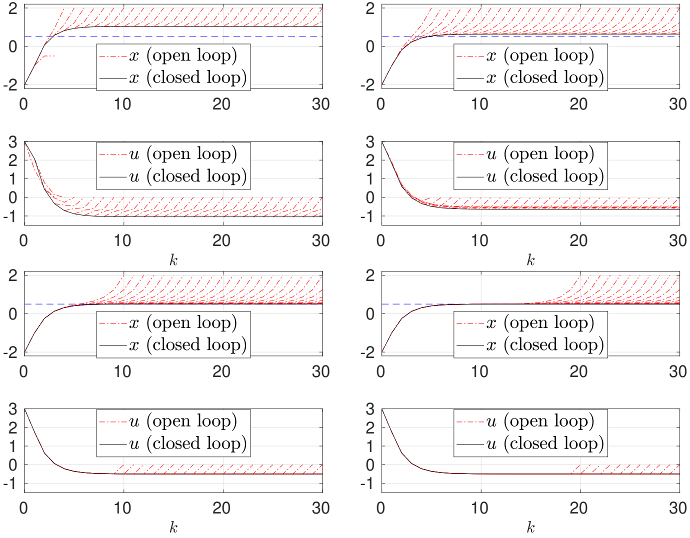


Turnpike property

- an approaching arc, converging to ≈ 0.5;
- a stable segment, staying at ≈ 0.5;
- a leaving arc, diverging from ≈ 0.5.

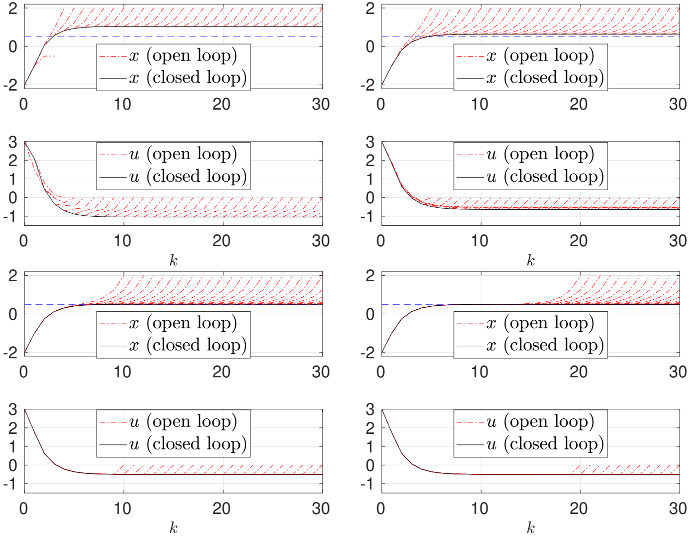
Model Predictive Control Schemes: Economic MPC

Open-loop and closed-loop solutions with respect to $N \in \{3, 5, 10, 20\}$



Model Predictive Control Schemes: Economic MPC

Open-loop and closed-loop solutions with respect to $N \in \{3, 5, 10, 20\}$



Optimal steady-state (x^e, u^e):
Can be calculated through

$$\min \; u^2 - x$$
$$\text{s.t. } x = x^+ = 2x + u$$

(but it is not necessary)

Alternatively, the running costs

$$\tilde{\ell}(x, u) = c_1|x - x^e|^2 + c_2|u - u^e|^2$$

asymptotically lead to the same closed-loop solution

However, the transient behavior is different

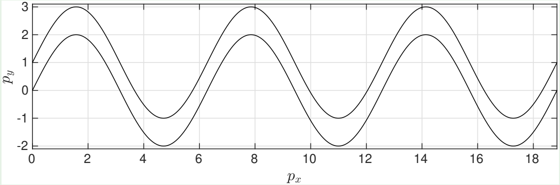
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \;\; p_y \leq 2\sin(p_x) + 1\}$$

Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0, 4]$, $w \in [-2, 2]$, $a \in [-4, 4]$ and $q \in [-6, 6]$.

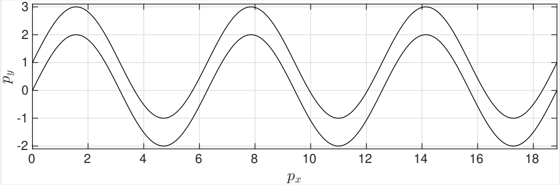
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \;\; p_y \leq 2\sin(p_x) + 1\}$$

Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0, 4]$, $w \in [-2, 2]$, $a \in [-4, 4]$ and $q \in [-6, 6]$.

Maximize the distance traveled in p_x-direction:

$$\max \int_0^{T_{end}} v(t)\cos(\phi(t))dt = -\min \int_0^{T_{end}} -v(t)\cos(\phi(t))dt$$

$$\max \Delta \sum_{i=0}^{K_{end}} v(i)\cos(\phi(i)) = -\min \Delta \sum_{i=0}^{K_{end}} -v(i)\cos(\phi(i))$$

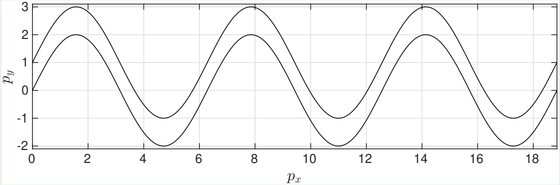
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

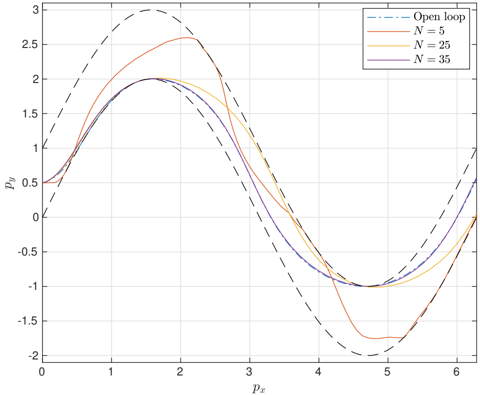
$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \;\; p_y \leq 2\sin(p_x) + 1\}$$

Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0, 4]$, $w \in [-2, 2]$, $a \in [-4, 4]$ and $q \in [-6, 6]$.

Maximize the distance traveled in p_x-direction:

$$\max \int_0^{T_{end}} v(t)\cos(\phi(t))dt = -\min \int_0^{T_{end}} -v(t)\cos(\phi(t))dt$$

$$\max \Delta \sum_{i=0}^{K_{end}} v(i)\cos(\phi(i)) = -\min \Delta \sum_{i=0}^{K_{end}} -v(i)\cos(\phi(i))$$



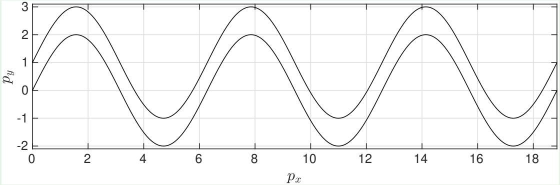
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

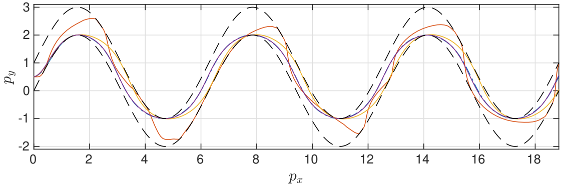
$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \;\; p_y \leq 2\sin(p_x) + 1\}$$

Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0,4]$, $w \in [-2,2]$, $a \in [-4,4]$ and $q \in [-6,6]$.

Maximize the distance traveled in p_x-direction:

$$\max \int_0^{T_{end}} v(t)\cos(\phi(t))dt = -\min \int_0^{T_{end}} -v(t)\cos(\phi(t))dt$$

$$\max \Delta \sum_{i=0}^{K_{end}} v(i)\cos(\phi(i)) = -\min \Delta \sum_{i=0}^{K_{end}} -v(i)\cos(\phi(i))$$



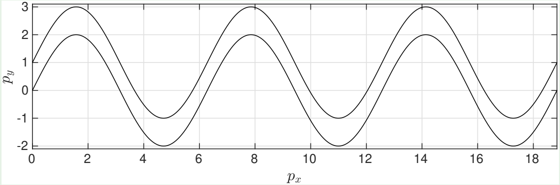
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

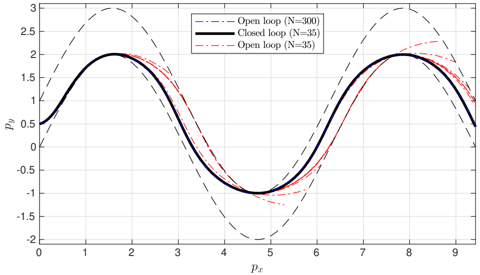
$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \ \ p_y \leq 2\sin(p_x) + 1\}$$

Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0, 4]$, $w \in [-2, 2]$, $a \in [-4, 4]$ and $q \in [-6, 6]$.

Maximize the distance traveled in p_x-direction:

$$\max \int_0^{T_{end}} v(t)\cos(\phi(t))dt = -\min \int_0^{T_{end}} -v(t)\cos(\phi(t))dt$$

$$\max \Delta \sum_{i=0}^{K_{end}} v(i)\cos(\phi(i)) = -\min \Delta \sum_{i=0}^{K_{end}} -v(i)\cos(\phi(i))$$



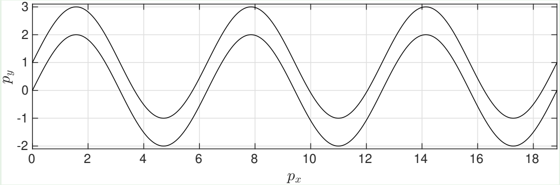
Model Predictive Control Schemes: Economic MPC

Example (Mobile Robot)

Continuous and discrete time dynamics ($\Delta > 0$):

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ w \\ a \\ q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p_x^+ \\ p_y^+ \\ \phi^+ \\ v^+ \\ w^+ \end{bmatrix} = \begin{bmatrix} p_x + \Delta v\cos(\phi) \\ p_y + \Delta v\sin(\phi) \\ \phi + \Delta w \\ v + \Delta a \\ w + \Delta q \end{bmatrix}$$

Goal: Finish the "race track" as quick as possible



The track is defined through the set

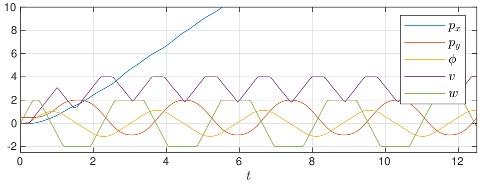
$$\mathcal{P} = \{[p_x, p_y]^T \in \mathbb{R}^2 | p_y \geq 2\sin(p_x), \;\; p_y \leq 2\sin(p_x) + 1\}$$

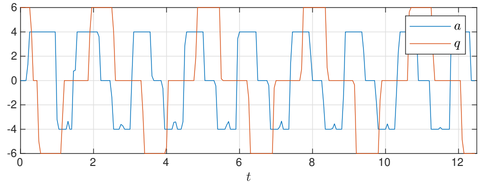
Additional constraints: $p_x(t_1) \leq p_x(t_2)$ for all $t_1 \leq t_2$.
$v \in [0, 4]$, $w \in [-2, 2]$, $a \in [-4, 4]$ and $q \in [-6, 6]$.

Maximize the distance traveled in p_x-direction:

$$\max \int_0^{T_{end}} v(t)\cos(\phi(t))dt = -\min \int_0^{T_{end}} -v(t)\cos(\phi(t))dt$$

$$\max \Delta \sum_{i=0}^{K_{end}} v(i)\cos(\phi(i)) = -\min \Delta \sum_{i=0}^{K_{end}} -v(i)\cos(\phi(i))$$



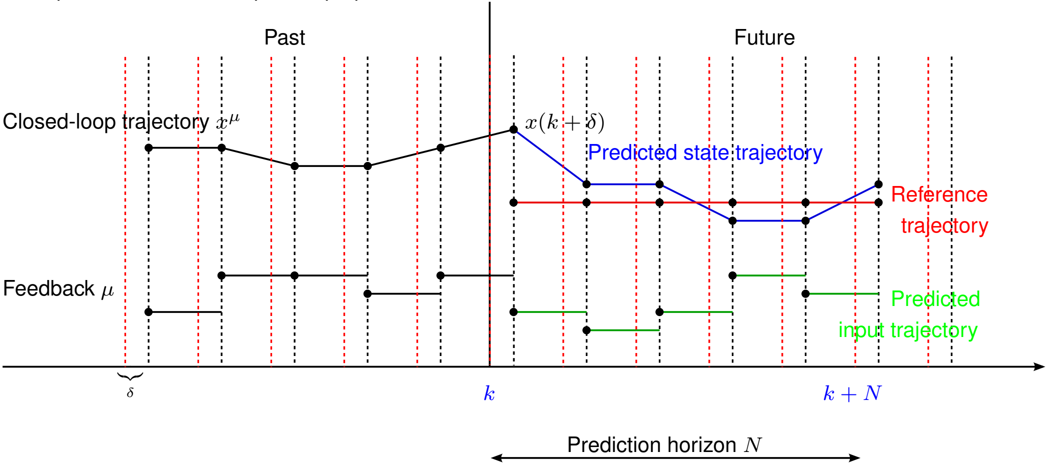


Section 3

Implementational Aspects of MPC

Implementational Aspects of MPC

So far, we have implicitly assumed that the OCP can be solved instantaneously
$\rightsquigarrow$ Introduce $\delta > 0$ as an upper bound for the time to solve the OCP, or
$\rightsquigarrow$ use multiple elements of the open-loop optimal solution to define the feedback law



Warm-Start

Optimal open loop input and state trajectories at time $k \in \mathbb{N}$ with respect to the initial state x_0:

$$u^\star(\cdot; k, x_0) = \begin{bmatrix} u_N^\star(0; k, x_0) \\ \vdots \\ u_N^\star(N-1; k, x_0) \end{bmatrix}, \qquad x^\star(\cdot; k, x_0) = \begin{bmatrix} x_N^\star(0; k, x_0) \\ \vdots \\ x_N^\star(N-1; k, x_0) \end{bmatrix}$$

Warm-Start

Optimal open loop input and state trajectories at time $k \in \mathbb{N}$ with respect to the initial state x_0:

$$u^\star(\cdot; k, x_0) = \begin{bmatrix} u_N^\star(0; k, x_0) \\ \vdots \\ u_N^\star(N-1; k, x_0) \end{bmatrix}, \qquad x^\star(\cdot; k, x_0) = \begin{bmatrix} x_N^\star(0; k, x_0) \\ \vdots \\ x_N^\star(N-1; k, x_0) \end{bmatrix}$$

If the prediction horizon N is large, it is not unreasonable to assume that

$$u^\star(i+1; k, x_0) \approx u^\star(i; k+1, x^\star(1; k, x_0)), \qquad x^\star(i+1; k, x_0) \approx x^\star(i; k+1, x^\star(1; k, x_0))$$

is satisfied for $i = 0, \dots, N-2$.

Warm-Start

Optimal open loop input and state trajectories at time $k \in \mathbb{N}$ with respect to the initial state x_0:

$$u^\star(\cdot; k, x_0) = \begin{bmatrix} u_N^\star(0; k, x_0) \\ \vdots \\ u_N^\star(N-1; k, x_0) \end{bmatrix}, \qquad x^\star(\cdot; k, x_0) = \begin{bmatrix} x_N^\star(0; k, x_0) \\ \vdots \\ x_N^\star(N-1; k, x_0) \end{bmatrix}$$

If the prediction horizon N is large, it is not unreasonable to assume that

$$u^\star(i+1; k, x_0) \approx u^\star(i; k+1, x^\star(1; k, x_0)), \qquad x^\star(i+1; k, x_0) \approx x^\star(i; k+1, x^\star(1; k, x_0))$$

is satisfied for $i = 0, \ldots, N-2$.

The initialization

$$u^0(\cdot; k+1, x^\star(1; k, x_0)) = \begin{bmatrix} u_N^\star(1; k, x_0) \\ \vdots \\ u_N^\star(N-1; k, x_0) \\ 0 \end{bmatrix} \qquad x^0(\cdot; k+1, x^\star(1; k, x_0)) = \begin{bmatrix} x_N^\star(1; k, x_0) \\ \vdots \\ x_N^\star(N-1; k, x_0) \\ f(x_N^\star(N-1; k, x_0), 0) \end{bmatrix}$$

CAN reduce the numerical complexity at the next time step $k+1$ significantly.

Warm-Start

Optimal open loop input and state trajectories at time $k \in \mathbb{N}$ with respect to the initial state x_0:

$$u^\star(\cdot;k,x_0) = \begin{bmatrix} u_N^\star(0;k,x_0) \\ \vdots \\ u_N^\star(N-1;k,x_0) \end{bmatrix}, \qquad x^\star(\cdot;k,x_0) = \begin{bmatrix} x_N^\star(0;k,x_0) \\ \vdots \\ x_N^\star(N-1;k,x_0) \end{bmatrix}$$

If the prediction horizon N is large, it is not unreasonable to assume that

$$u^\star(i+1;k,x_0) \approx u^\star(i;k+1,x^\star(1;k,x_0)), \qquad x^\star(i+1;k,x_0) \approx x^\star(i;k+1,x^\star(1;k,x_0))$$

is satisfied for $i = 0, \ldots, N-2$.

The initialization

$$u^0(\cdot;k+1,x^\star(1;k,x_0)) = \begin{bmatrix} u_N^\star(1;k,x_0) \\ \vdots \\ u_N^\star(N-1;k,x_0) \\ 0 \end{bmatrix} \qquad x^0(\cdot;k+1,x^\star(1;k,x_0)) = \begin{bmatrix} x_N^\star(1;k,x_0) \\ \vdots \\ x_N^\star(N-1;k,x_0) \\ f(x_N^\star(N-1;k,x_0),0) \end{bmatrix}$$

CAN reduce the numerical complexity at the next time step $k+1$ significantly.

Remark

If the OCP is non-convex and has multiple local minima, warm-start may be counterproductive in finding a global minimum.

Formulation of the Optimization Problem

Standard formulation of an optimization problem:

$$y^\star = \arg \min_{y \in \mathbb{R}^{\alpha_1}} \; F(y)$$

$$\begin{aligned} \text{s.t.} \quad & G_i(y) \leq 0, \qquad & i = 1, \ldots, \alpha_2 \\ & H_j(y) = 0, \qquad & j = 1, \ldots, \alpha_3 \end{aligned}$$

Different possibilities to define the unknown y:
Option 1: (Full discretization)

$$y = \begin{bmatrix} x(0)^T & u(0)^T & \cdots & x(N-1)^T & u(N-1)^T & x(N)^T \end{bmatrix}^T.$$

- More unknowns, larger number of constraints, but sparsity patterns

Option 2: (Recursive elimination)

$$y = \begin{bmatrix} u(0)^T & \cdots & u(N-1)^T \end{bmatrix}^T,$$

- Smaller number of unknowns, smaller number of constraints, but dense representations

Also see:

- Single shooting & multiple shooting

(Depending on the optimization algorithm, different representations have advantages/disadvantages)

Introduction to Nonlinear Control

Stability, control design, and estimation

Philipp Braun & Christopher M. Kellett
School of Engineering,
Australian National University, Canberra, Australia

Part II:

Chapter 15: Model Predictive Control

Australian National University