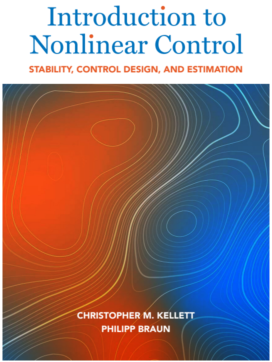
Introduction to Nonlinear Control

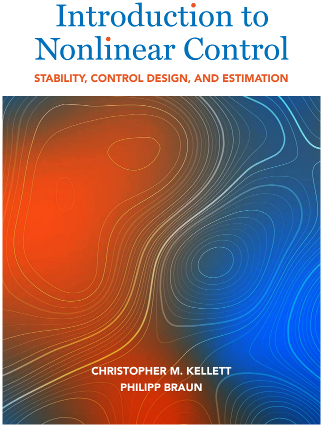
Stability, control design, and estimation

Christopher M. Kellett & Philipp Braun

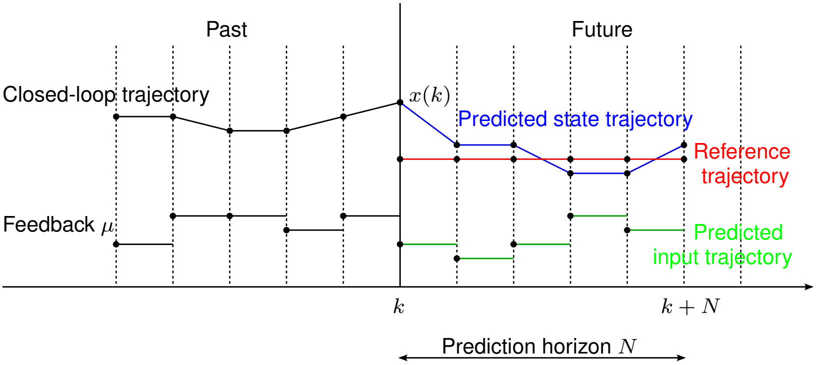
PRINCETON UNIVERSITY PRESS



Part II: Controller Design



Model Predictive Control & the Receding Horizon Principle



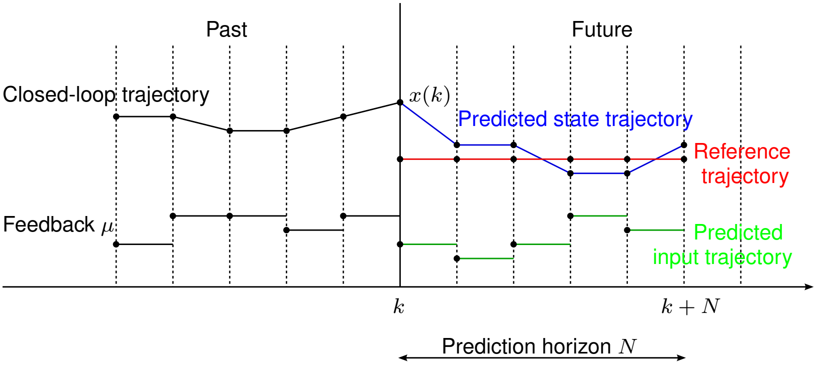
We consider discrete time systems

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

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 & the Receding Horizon Principle



We consider discrete time systems

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

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

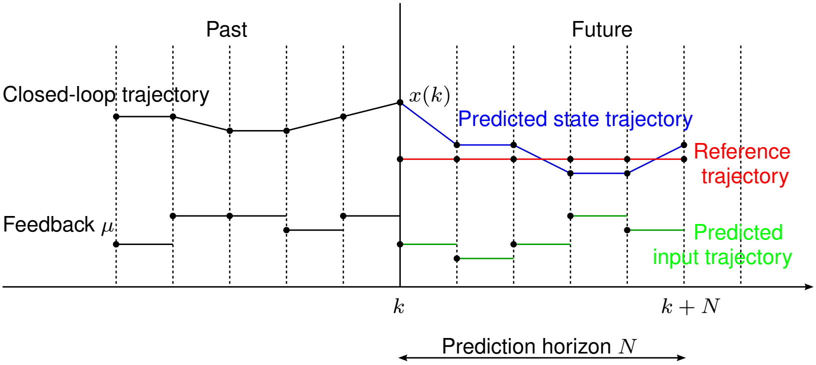
- State constraints $x \in \mathbb{X} \subset \mathbb{R}^n$
- Input constraints $u \in \mathbb{U}(x) \subset \mathbb{R}^m$

- 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 & the Receding Horizon Principle



MPC is also known as

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

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 $(0,0) \in \mathbb{D}$

The Basic MPC Algorithm

- 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 Algorithm

- 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 Algorithm

- 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 Algorithm

- 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 (OCP)

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

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

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

The Basic MPC Algorithm

- 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 (OCP)

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

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

  ($\rightsquigarrow$ 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 Algorithm

- 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 (OCP)

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

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

($\rightsquigarrow$ 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 Algorithm

- 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 (OCP)

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

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

($\rightsquigarrow$ 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 Algorithm

- 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 (OCP)

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

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

($\rightsquigarrow$ 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 Algorithm (Part 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 $\mathbb{X}_F \subset \mathbb{R}^n$.

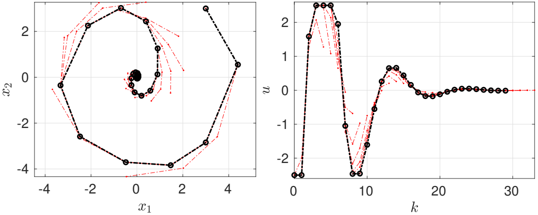
For $k = 0, 1, 2, \ldots$

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.

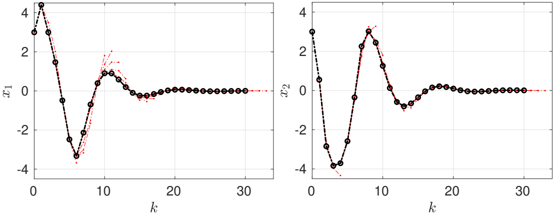
MPC Example

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: $F(x) = x^T x$, $\mathbb{X}_F = \mathbb{R}^2$.

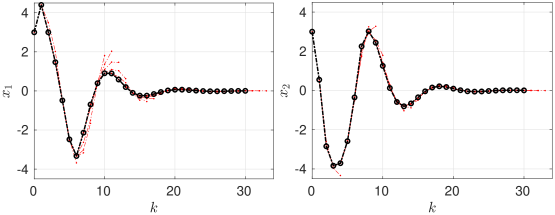


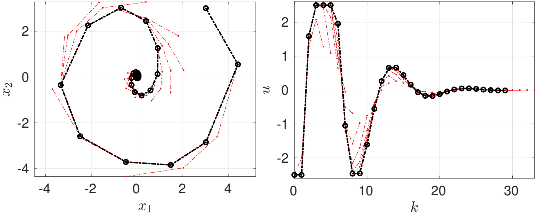
MPC Example

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

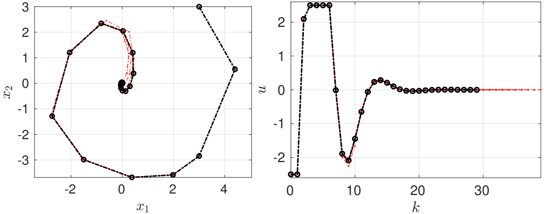
$$A = \begin{bmatrix} \frac{6}{5} & \frac{6}{5} \\ -\frac{1}{2} & \frac{6}{5} \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$$

- 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 $x_0 = [3\ 3]^T$)



MPC Closed-Loop Analysis (Discussion)

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 (Discussion)

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 (Discussion)

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 (Discussion)

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 (Discussion)

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 (Discussion)

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 (Discussion)

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

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: ($Q \geq 0$, $R \geq 0$):

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

$\rightsquigarrow$ 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

$\rightsquigarrow$ 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: ($Q \geq 0$, $R \geq 0$):

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

$\rightsquigarrow$ 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

$\rightsquigarrow$ 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

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

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

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

$\rightsquigarrow$ 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 $x_0 \in \mathbb{X}$!

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 $k \in \mathbb{N}$.

$\rightsquigarrow$ 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*.

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

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

Definition (Recursive feasibility)

Consider the basic MPC Algorithm with

- input constraints $\mathbb{U}$
- set of initial states $\mathbb{X}_{\mathrm{rf}}^N \subset \mathbb{X}$

$\mathbb{X}_{\mathrm{rf}}^N$ is called recursively feasible if feasibility of the OCP for $x(0) = x_0 \in \mathbb{X}_{\mathrm{rf}}^N$ implies feasibility of the OCP for all $k \in \mathbb{N}$.

(Note that: Recursive feasibility depends on the prediction horizon $N \in \mathbb{N}$.)

Model Predictive Control Schemes in the literature

Model Predictive Control Schemes: (not a comprehensive list)

- MPC for Time-Varying Systems & Reference Tracking
- Linear MPC
- Nonlinar MPC
- MPC Without Terminal Costs & Terminal 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

Introduction to Nonlinear Control: Stability, control design, and estimation