CHALMERS

UNIVERSITY OF TECHNOLOGY

SSY281 - MODEL PREDICTIVE CONTROL

NIKOLCE MURGOVSKI

Division of Systems and Control
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden

2021-02-26

Lecture 9 : Feasibility

Goals for today:

- To understand the issue of recursive feasibility and how to obtain it
- To understand the role of constraint management and back-up strategies

Lecture 9 : Feasibility

Goals for today:

- To understand the issue of recursive feasibility and how to obtain it
- To understand the role of constraint management and back-up strategies

Learning objectives:

- Understand and explain the basic principles of model predictive control, its pros and cons, and the challenges met in implementation and applications
- Describe basic properties of MPC controllers and analyse algorithmic details on very simple examples
- Understand and explain basic properties of the optimisation problem as an ingredient of MPC, in particular concepts like linear, quadratic and convex optimisation, optimality conditions, and feasibility

Recursive (persistent) feasibility – definition

Definition (Invariant set)

The set $\mathcal{S}$ is *positively invariant* for the autonomous system $x^+ = f_a(x)$ if

$$x(0) \in \mathcal{S} \quad \Rightarrow \quad x(k) \in \mathcal{S}, \quad \forall k \in \mathbb{N}_+.$$

Recursive (persistent) feasibility – definition

Definition (Invariant set)

The set $\mathcal{S}$ is *positively invariant* for the autonomous system $x^+ = f_a(x)$ if

$$x(0) \in \mathcal{S} \quad \Rightarrow \quad x(k) \in \mathcal{S}, \quad \forall k \in \mathbb{N}_+.$$

Definition (Recursive (persistent) feasibility)

The receding horizon controller is recursively (persistently) feasible if the feasible set $\mathcal{X}_N$ is positively invariant for the closed-loop system $x^+ = f(x, \kappa_N(x))$, i.e.

$$x(0) \in \mathcal{X}_N \quad \Rightarrow \quad x(k+1) = f(x(k), \kappa_N(x(k))) \in \mathcal{X}_N, \quad \forall k \in \mathbb{N}_+.$$

Recursive (persistent) feasibility – definition

Definition (Invariant set)

The set $\mathcal{S}$ is *positively invariant* for the autonomous system $x^+ = f_a(x)$ if

$$x(0) \in \mathcal{S} \quad \Rightarrow \quad x(k) \in \mathcal{S}, \quad \forall k \in \mathbb{N}_+.$$

Definition (Recursive (persistent) feasibility)

The receding horizon controller is recursively (persistently) feasible if the feasible set $\mathcal{X}_N$ is positively invariant for the closed-loop system $x^+ = f(x, \kappa_N(x))$, i.e.

$$x(0) \in \mathcal{X}_N \quad \Rightarrow \quad x(k+1) = f(x(k), \kappa_N(x(k))) \in \mathcal{X}_N, \quad \forall k \in \mathbb{N}_+.$$

The implication of these definitions is that if the RHC is recursively feasible, then it is guaranteed that the optimisation problem to be solved at every time instant is feasible, provided the initial state is feasible.

Example: feasibility vs. recursive feasibility

Consider the system

$$x(k+1) = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k)$$

with constraints $x_1(k) \in [0, 10]$, $u(k) \in [-1, 1], \forall k$. We will study the finite-time optimal control problem with

$$N = 2, \quad x(0) = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad Q = P_f = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1.$$

Example: feasibility vs. recursive feasibility

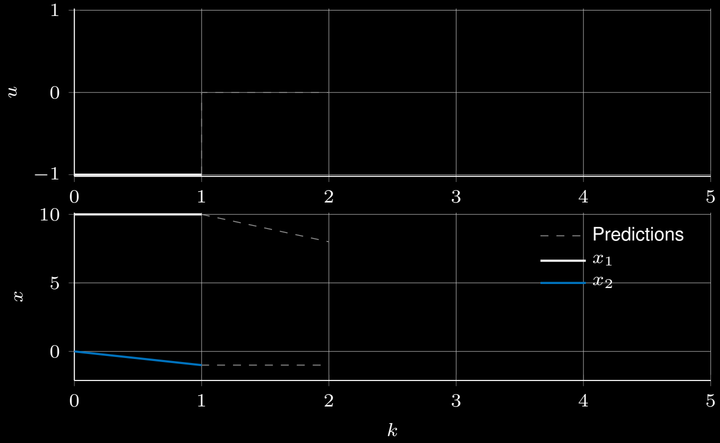

Figure 32: Evolutional of states and control actions. The controller is feasible over the first 3 samples and MPC predicts that the 1st state will reach zero at the 4th sample. However, after evaluating the controller at the 3rd sample, it is clear that even the highest control action, depicted by the dashed line, is not able to keep the 1st state above zero.

Example: feasibility vs. recursive feasibility

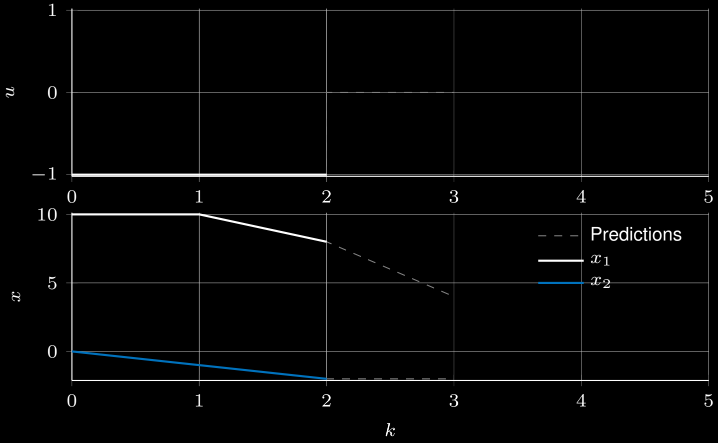

Figure 32: Evolutional of states and control actions. The controller is feasible over the first 3 samples and MPC predicts that the 1st state will reach zero at the 4th sample. However, after evaluating the controller at the 3rd sample, it is clear that even the highest control action, depicted by the dashed line, is not able to keep the 1st state above zero.

Example: feasibility vs. recursive feasibility

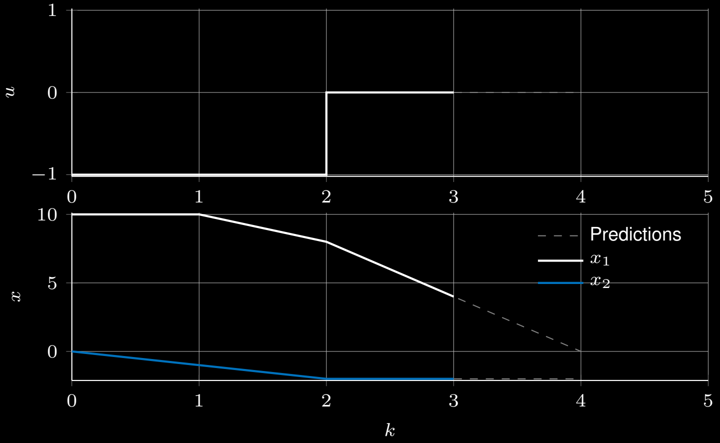

Figure 32: Evolutional of states and control actions. The controller is feasible over the first 3 samples and MPC predicts that the 1st state will reach zero at the 4th sample. However, after evaluating the controller at the 3rd sample, it is clear that even the highest control action, depicted by the dashed line, is not able to keep the 1st state above zero.

Example: feasibility vs. recursive feasibility

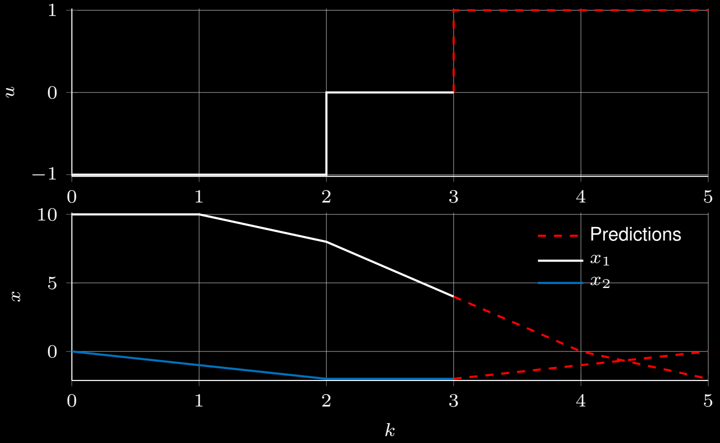

Figure 32: Evolutional of states and control actions. The controller is feasible over the first 3 samples and MPC predicts that the 1st state will reach zero at the 4th sample. However, after evaluating the controller at the 3rd sample, it is clear that even the highest control action, depicted by the dashed line, is not able to keep the 1st state above zero.

Example: feasibility vs. recursive feasibility

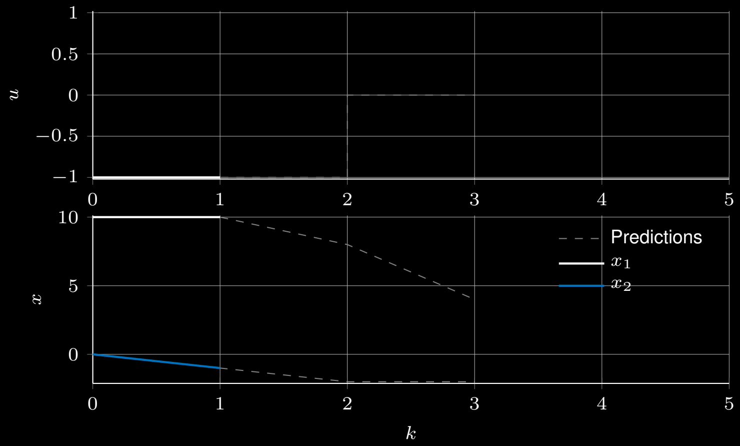

Figure 33: Evolutional of states and control actions. For a prediction horizon of $N = 3$ the controller is recursively feasible for the given initial state.

Example: feasibility vs. recursive feasibility

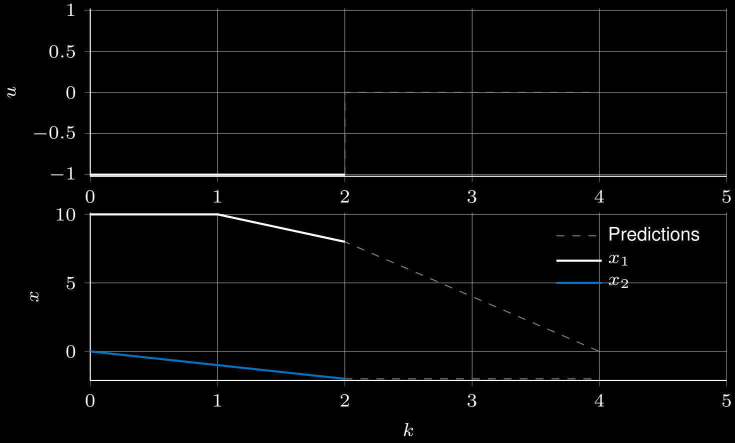

Figure 33: Evolutional of states and control actions. For a prediction horizon of $N = 3$ the controller is recursively feasible for the given initial state.

Example: feasibility vs. recursive feasibility

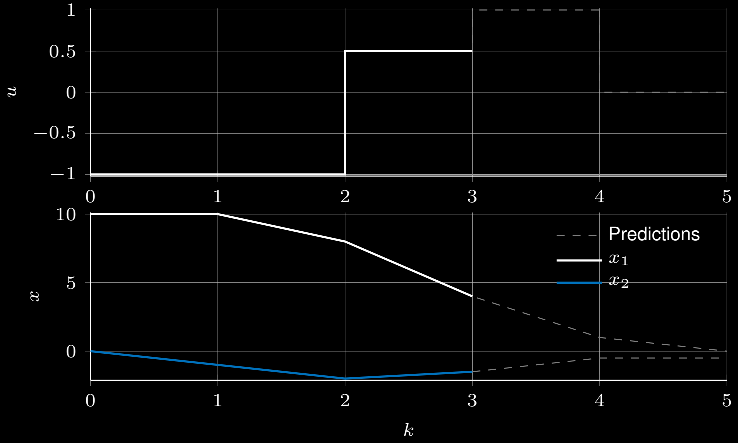

Figure 33: Evolutional of states and control actions. For a prediction horizon of $N = 3$ the controller is recursively feasible for the given initial state.

Example: feasibility vs. recursive feasibility

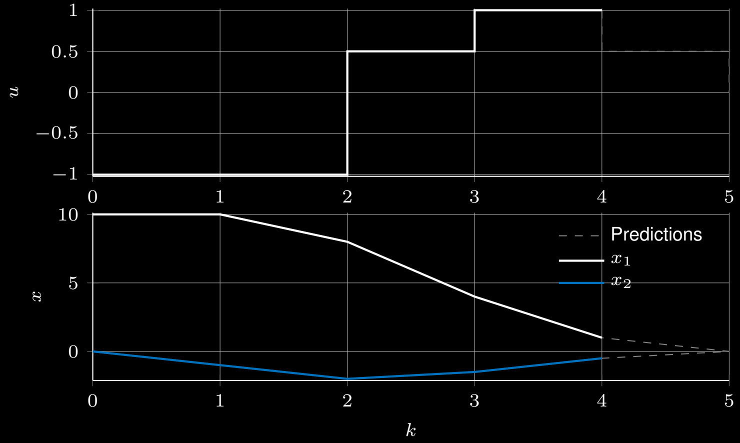

Figure 33: Evolutional of states and control actions. For a prediction horizon of $N = 3$ the controller is recursively feasible for the given initial state.

Illustration of recursive feasibility

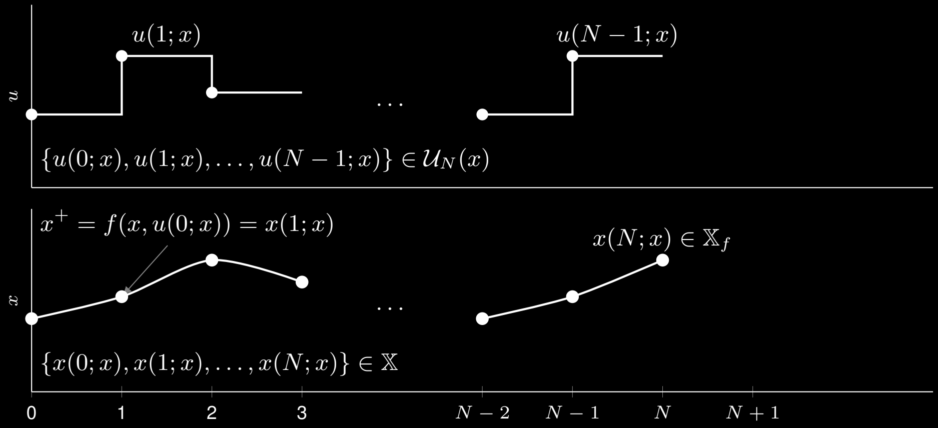

Figure 34: Illustration of recursive feasibility. After shifting the horizon and applying a control sequence constructed from the tail of the previous sequence and a new feasible control u, recursive feasibility requires that the new state sequence is also feasible.

Illustration of recursive feasibility

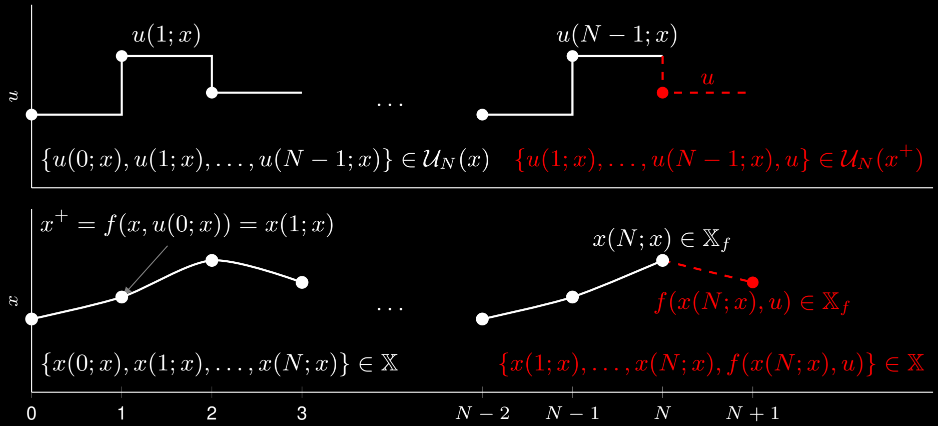

Figure 34: Illustration of recursive feasibility. After shifting the horizon and applying a control sequence constructed from the tail of the previous sequence and a new feasible control u, recursive feasibility requires that the new state sequence is also feasible.

Condition for recursive feasibility

Definition (Control invariant set)

A set $\mathcal{C} \subseteq \mathbb{X}$ is a *control invariant set* of the system $x^+ = f(x,u)$ if

$$x \in \mathcal{C} \quad \Rightarrow \quad \exists u \in \mathbb{U} \text{ such that } x^+ = f(x,u) \in \mathcal{C}.$$

The *maximal control invariant set* contained in $\mathbb{X}$ is denoted $\mathcal{C}_\infty$ and contains all control invariant sets in $\mathbb{X}$.

Condition for recursive feasibility

Definition (Control invariant set)

A set $\mathcal{C} \subseteq \mathbb{X}$ is a *control invariant set* of the system $x^+ = f(x,u)$ if

$$x \in \mathcal{C} \quad \Rightarrow \quad \exists u \in \mathbb{U} \text{ such that } x^+ = f(x,u) \in \mathcal{C}.$$

The *maximal control invariant set* contained in $\mathbb{X}$ is denoted $\mathcal{C}_\infty$ and contains all control invariant sets in $\mathbb{X}$.

Theorem (Sufficient condition for recursive feasibility)

The receding horizon controller based on the finite horizon optimal control problem (37)-(40) is recursively feasible if the terminal constraint set $\mathbb{X}_f$ is control invariant.

Computation of the maximal control invariant set

The following recursion can be used to find $\mathcal{C}_\infty$:

$$
\begin{aligned}
\Omega_0 &= \mathbb{X} \\
\Omega_{i+1} &= \text{Pre}\,(\Omega_i) \cap \Omega_i .
\end{aligned} \tag{83}
$$

Here, Pre $(\mathcal{S})$ is the set of states that can be driven into the target set $\mathcal{S}$ in one time step:

$$
\text{Pre}\,(\mathcal{S}) = \{x \in \mathbb{X} \mid \exists u \in \mathbb{U} \text{ such that } x^+ = f(x,u) \in \mathcal{S}\}. \tag{84}
$$

Computation of the maximal control invariant set

The following recursion can be used to find $\mathcal{C}_\infty$:

$$\begin{aligned} \Omega_0 &= \mathbb{X} \\ \Omega_{i+1} &= \text{Pre}\left(\Omega_i\right) \cap \Omega_i. \end{aligned} \tag{83}$$

Here, $\text{Pre}\left(\mathcal{S}\right)$ is the set of states that can be driven into the target set $\mathcal{S}$ in one time step:

$$\text{Pre}\left(\mathcal{S}\right) = \{x \in \mathbb{X} \mid \exists u \in \mathbb{U} \text{ such that } x^+ = f(x,u) \in \mathcal{S}\}. \tag{84}$$

The recursion (83), which generates a sequence of decreasing sets, may or may not terminate; if it does, $\Omega_{i+1} = \Omega_i$ for some i, the *determinedness index* of $\mathcal{C}_\infty$, which is in this case *finitely determined*. Figure 35 illustrates the different sets we have defined.

Illustration of a maximal control invariant set

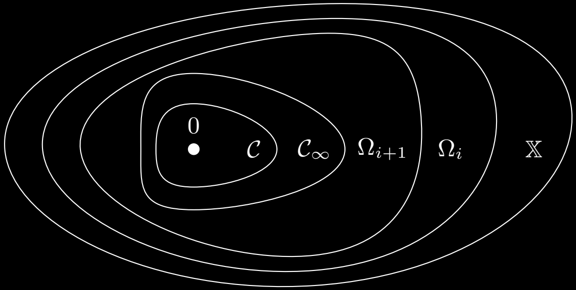

Figure 35: Construction of a maximal control invariant set $\mathcal{C}_\infty$ in $\mathbb{X}$. $\mathcal{C}$ is an arbitrary control invariant set.

Continuation of the previous example: maximal control invariant set

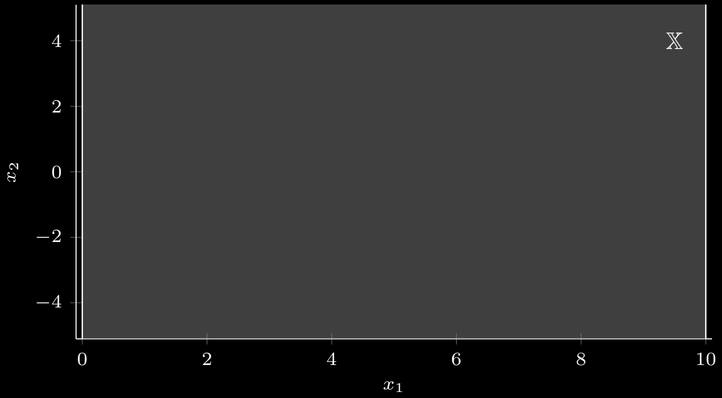

Figure 36: Maximal control invariant set reached with 3 recursions of Pre $(\Omega_i) \cap \Omega_i$.

Continuation of the previous example: maximal control invariant set

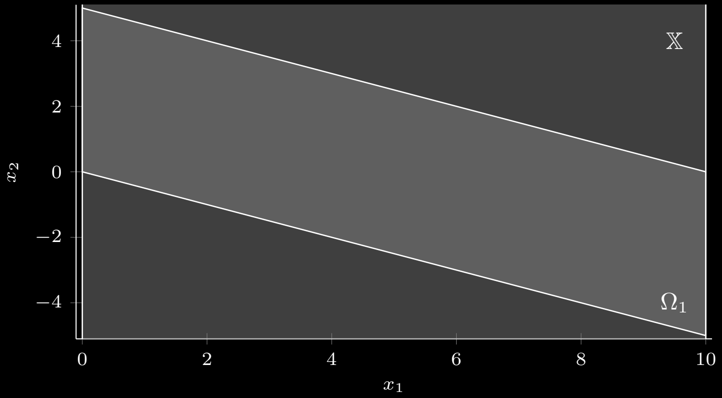

Figure 36: Maximal control invariant set reached with 3 recursions of Pre $(\Omega_i) \cap \Omega_i$.

Continuation of the previous example: maximal control invariant set

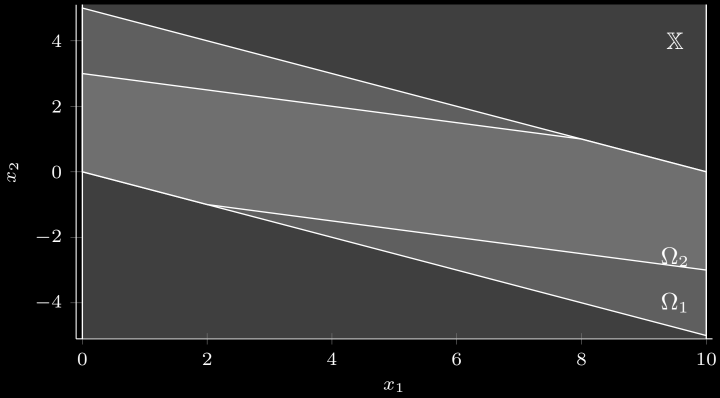

Figure 36: Maximal control invariant set reached with 3 recursions of Pre $(\Omega_i) \cap \Omega_i$.

Continuation of the previous example: maximal control invariant set

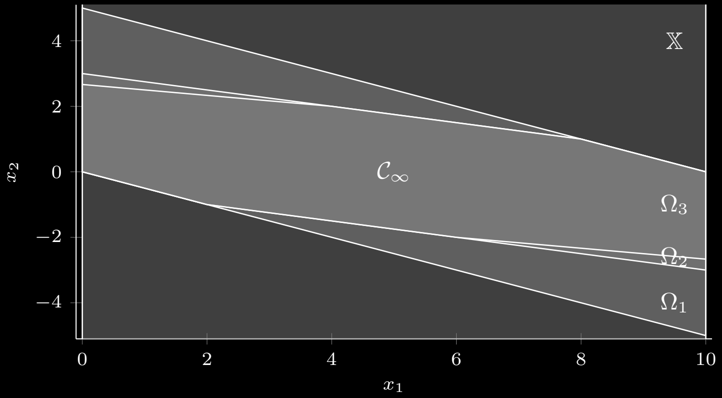

Figure 36: Maximal control invariant set reached with 3 recursions of Pre $(\Omega_i) \cap \Omega_i$.

Continuation of the previous example: feasible trajectories

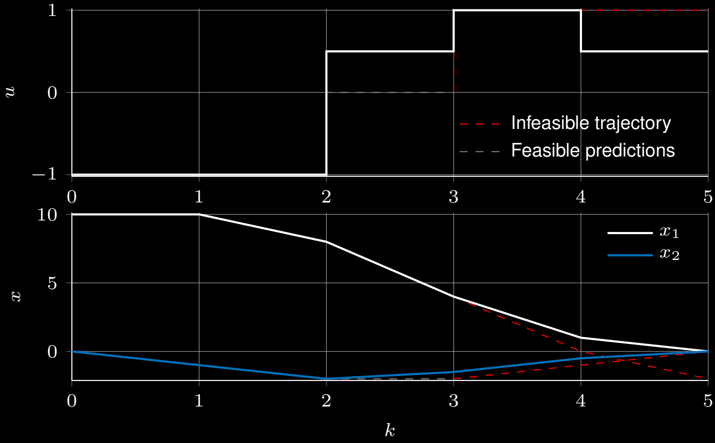

Figure 37: Comparison between the infeasible trajectories, depicted by dashed lines, and recursively feasible trajectories, solid lines, obtained when the target set is the maximal control invariant set.

i-step controllable set

Definition (i-step controllable set $\mathcal{K}_i(\mathcal{S})$)

The *i-step controllable set* $\mathcal{K}_i(\mathcal{S})$ is defined as the set of initial states that can be driven to the target set $\mathcal{S}$ in i steps, while satisfying state and control constraints at all times.

From the definition, it follows that the feasible set $\mathcal{X}_N$ can alternatively be defined as

$$\mathcal{X}_N = \mathcal{K}_N(\mathbb{X}_f).$$

i-step controllable set

Definition (i-step controllable set $\mathcal{K}_i(\mathcal{S})$)

The *i-step controllable set* $\mathcal{K}_i(\mathcal{S})$ is defined as the set of initial states that can be driven to the target set $\mathcal{S}$ in i steps, while satisfying state and control constraints at all times.

From the definition, it follows that the feasible set $\mathcal{X}_N$ can alternatively be defined as

$$\mathcal{X}_N = \mathcal{K}_N(\mathbb{X}_f).$$

If it is assumed that $\mathbb{X}_f$ is control invariant—which was shown above to guarantee persistent feasibility—then it is clear that the set sequence $\{\mathcal{K}_i(\mathbb{X}_f)\}$ is monotone in the sense

$$\mathbb{X}_f = \mathcal{K}_0(\mathbb{X}_f) \subseteq \ldots \subseteq \mathcal{K}_i(\mathbb{X}_f) \subseteq \mathcal{K}_{i+1}(\mathbb{X}_f) \subseteq \ldots \subseteq \mathcal{K}_N(\mathbb{X}_f) \subseteq \mathcal{K}_\infty(\mathbb{X}_f) \subseteq \mathbb{X}, \tag{85}$$

where $\mathcal{K}_\infty(\mathbb{X}_f)$ is the *maximal controllable set* with target set $\mathbb{X}_f$, defined by

$$\mathcal{K}_\infty(\mathbb{X}_f) = \bigcup_i \mathcal{K}_i(\mathbb{X}_f).$$

Illustration of an i-step controllable set

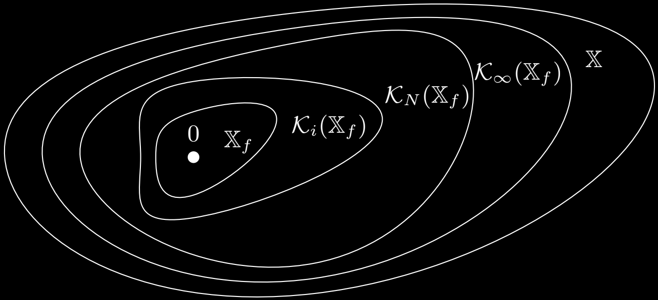

Figure 38: Construction of the maximal controllable set $\mathcal{K}_\infty(\mathbb{X}_f)$.

Continuation of the previous example: i-step controllable set

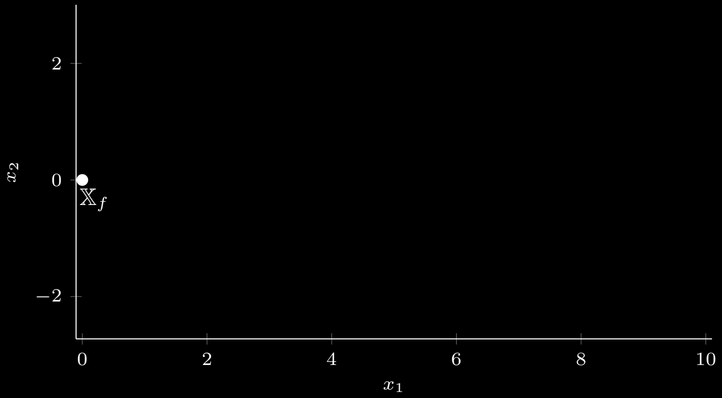

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix}10 & 0\end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

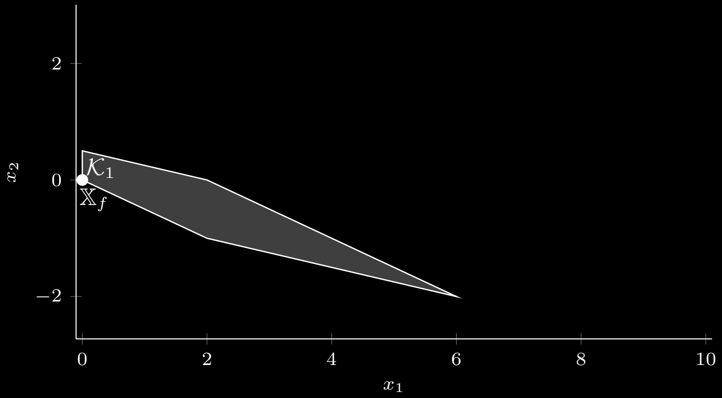

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix} 10 & 0 \end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

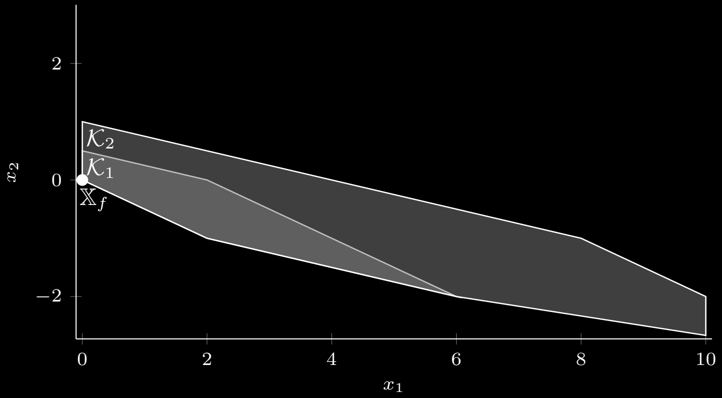

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix} 10 & 0 \end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

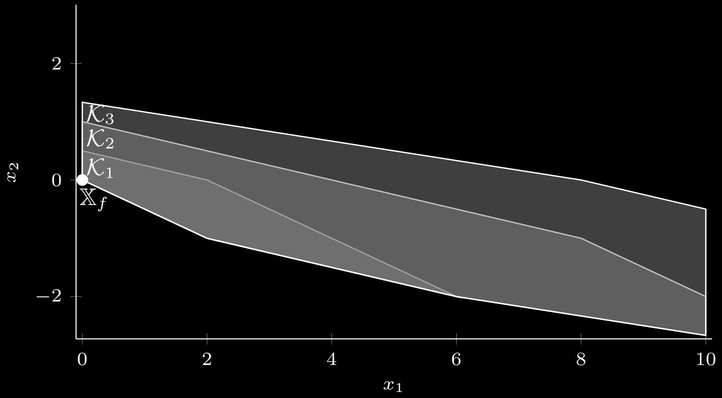

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix} 10 & 0 \end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

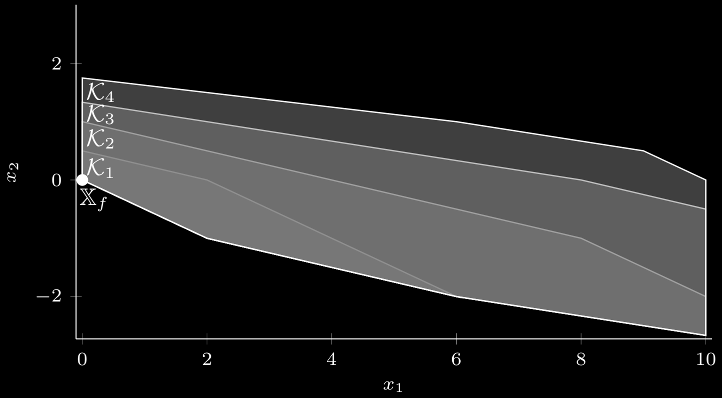

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix}10 & 0\end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

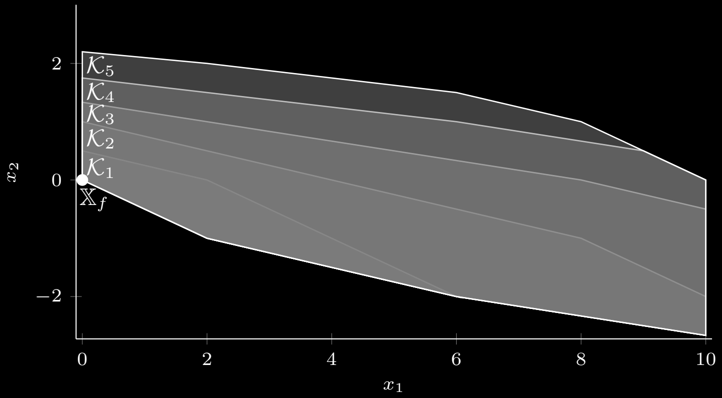

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $[10 \quad 0]^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

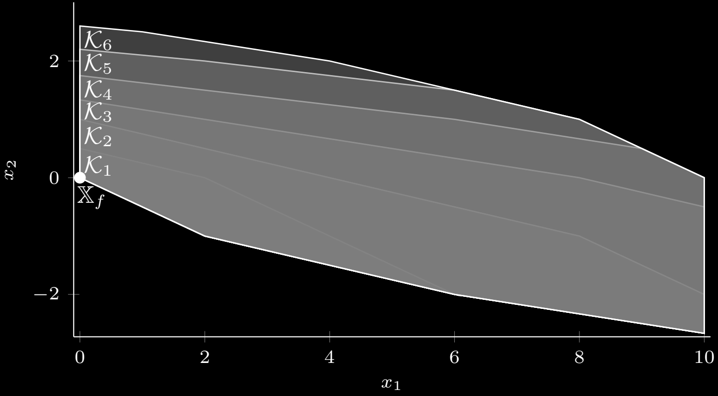

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix} 10 & 0 \end{bmatrix}^\top$ is persistently feasible.

Continuation of the previous example: i-step controllable set

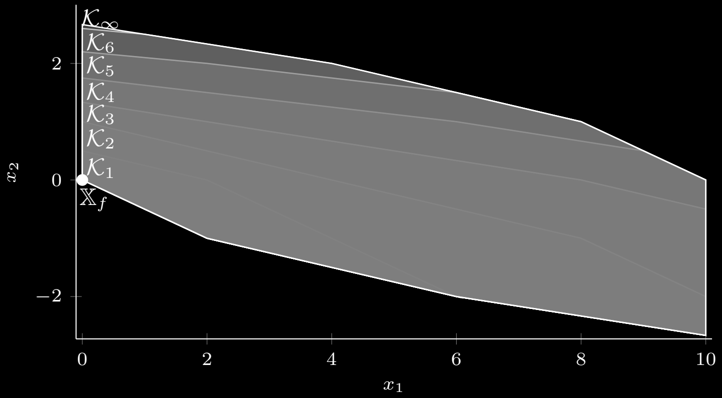

Figure 39: i-step controllable set for a target set $\mathbb{X}_f = \mathbf{0}$. For a horizon $N \geq 3$ the system initialised at $\begin{bmatrix}10 & 0\end{bmatrix}^\top$ is persistently feasible.

i-step controllable set with a larger target set

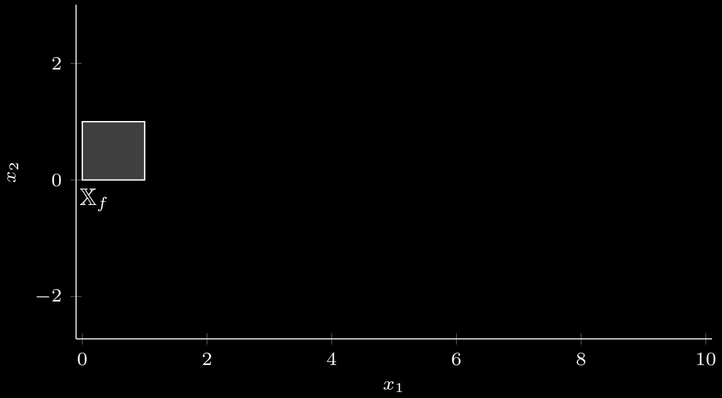

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

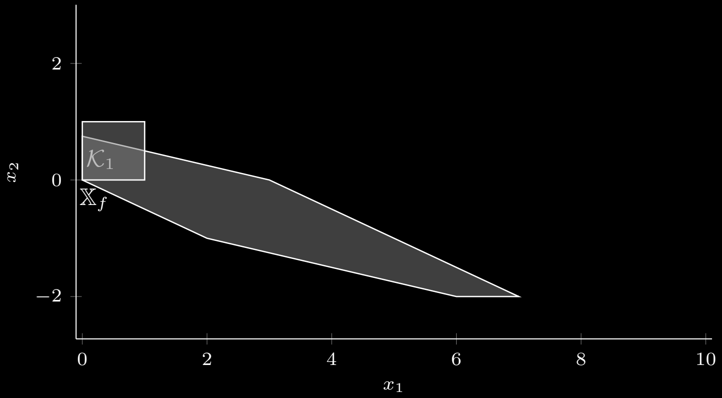

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

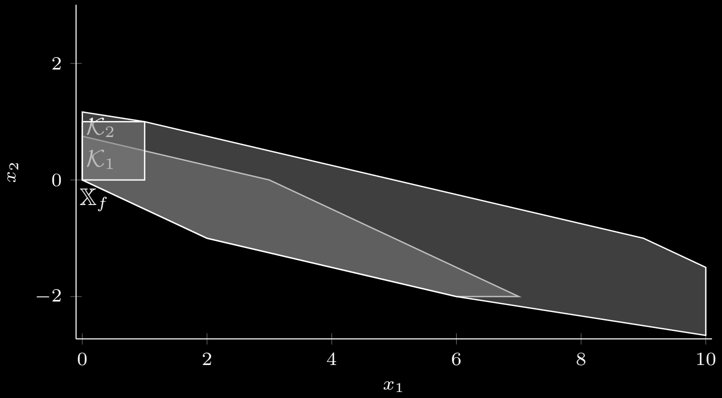

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

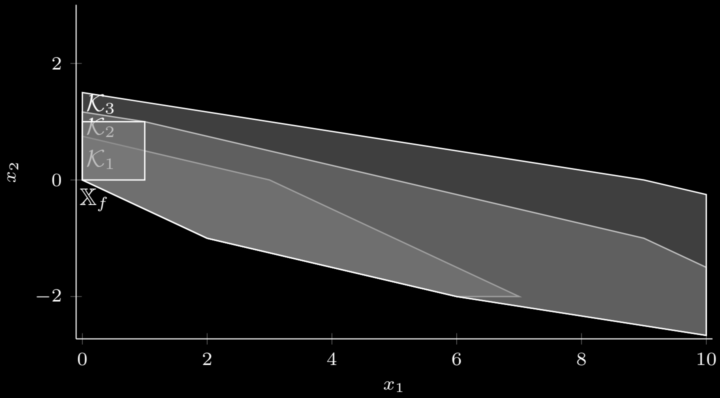

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x|0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

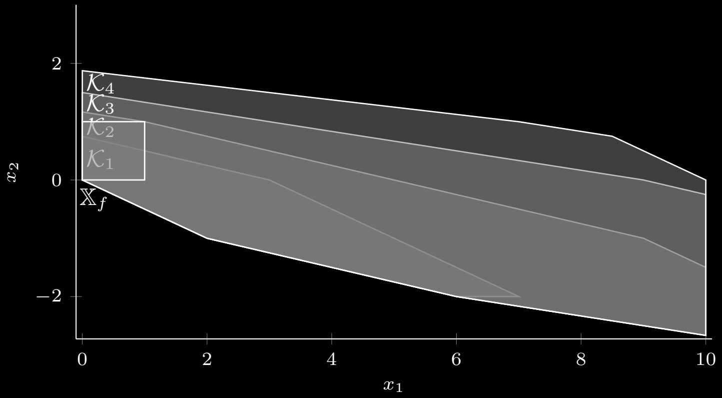

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

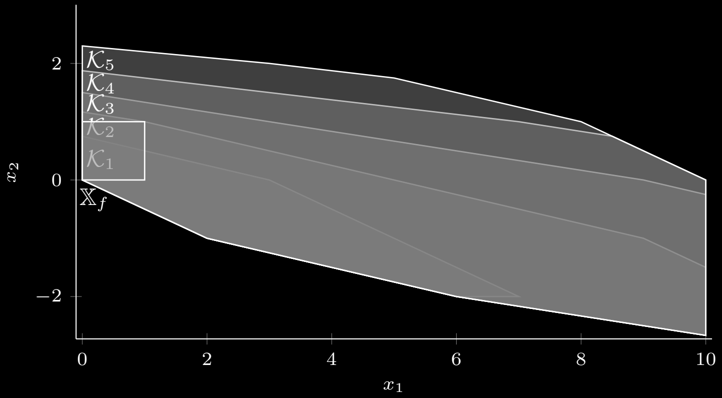

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

i-step controllable set with a larger target set

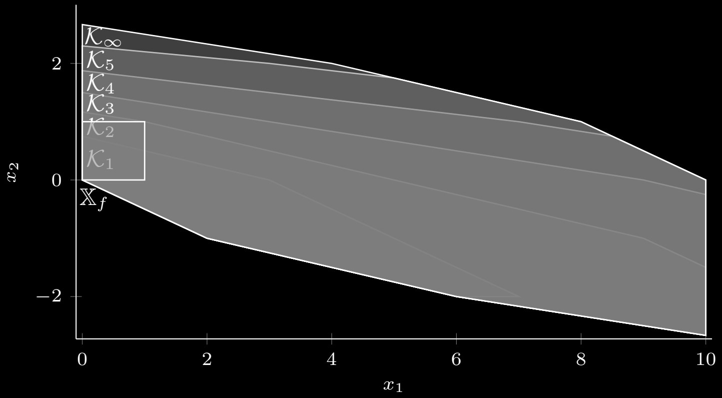

Figure 40: i-step controllable set for a larger target set $\mathbb{X}_f = \{x | 0 \leq x \leq 1\}$.

Constraint management

- Soft constraints can be introduced:

$$\min_{u} \quad V_N(\boldsymbol{u}) + ro\|\varepsilon\| \tag{86}$$

$$\text{subject to} \quad F\boldsymbol{u} + G\boldsymbol{x} \leq e + \varepsilon \tag{87}$$

$$\varepsilon \geq 0. \tag{88}$$

Constraint management

- Soft constraints can be introduced:

$$\min_{u} \quad V_N(\boldsymbol{u}) + ro\|\varepsilon\| \tag{86}$$

$$\text{subject to} \quad F\boldsymbol{u} + G\boldsymbol{x} \leq e + \varepsilon \tag{87}$$

$$\varepsilon \geq 0. \tag{88}$$

- Reduce window in which constraints are enforced.

Constraint management

- Soft constraints can be introduced:

$$\begin{aligned} \min_{u} \quad & V_N(\boldsymbol{u}) + ro\|\varepsilon\| && (86) \\ \text{subject to} \quad & F\boldsymbol{u} + G\boldsymbol{x} \leq e + \varepsilon && (87) \\ & \varepsilon \geq 0. && (88) \end{aligned}$$

- Reduce window in which constraints are enforced.
- Prioritise constraints $\rightarrow$ mixed-integer quadratic program.

Constraint management

- Soft constraints can be introduced:

$$\min_{u} \quad V_N(\boldsymbol{u}) + ro\|\varepsilon\| \tag{86}$$

$$\text{subject to} \quad F\boldsymbol{u} + G\boldsymbol{x} \leq e + \varepsilon \tag{87}$$

$$\varepsilon \geq 0. \tag{88}$$

- Reduce window in which constraints are enforced.
- Prioritise constraints $\rightarrow$ mixed-integer quadratic program.
- Don’t forget time limitations – control output *must* be delivered!

Feasibility – summary

- It may be impossible, for some initial states, to solve the RHC optimisation problem while respecting constraints on states and/or outputs. The problem is in these cases *infeasible*.

Feasibility – summary

- It may be impossible, for some initial states, to solve the RHC optimisation problem while respecting constraints on states and/or outputs. The problem is in these cases *infeasible*.
- In the nominal case, without disturbances and with a perfect model, *recursive feasibility* may be ensured by choosing the terminal constraint set $\mathbb{X}_f$ to be control invariant.

Feasibility – summary

- It may be impossible, for some initial states, to solve the RHC optimisation problem while respecting constraints on states and/or outputs. The problem is in these cases *infeasible*.
- In the nominal case, without disturbances and with a perfect model, *recursive feasibility* may be ensured by choosing the terminal constraint set $\mathbb{X}_f$ to be control invariant.
- In practice, infeasibility may still occur, possibly caused by too strict performance requirements, a large disturbance, model uncertainty, or by an unstable system.

Feasibility – summary

- It may be impossible, for some initial states, to solve the RHC optimisation problem while respecting constraints on states and/or outputs. The problem is in these cases *infeasible*.
- In the nominal case, without disturbances and with a perfect model, *recursive feasibility* may be ensured by choosing the terminal constraint set $\mathbb{X}_f$ to be control invariant.
- In practice, infeasibility may still occur, possibly caused by too strict performance requirements, a large disturbance, model uncertainty, or by an unstable system.
- Ad hoc solutions are for example to keep the control as is (from the previous sampling instant), or to switch to a backup controller.

Feasibility – summary

- It may be impossible, for some initial states, to solve the RHC optimisation problem while respecting constraints on states and/or outputs. The problem is in these cases *infeasible*.
- In the nominal case, without disturbances and with a perfect model, *recursive feasibility* may be ensured by choosing the terminal constraint set $\mathbb{X}_f$ to be control invariant.
- In practice, infeasibility may still occur, possibly caused by too strict performance requirements, a large disturbance, model uncertainty, or by an unstable system.
- Ad hoc solutions are for example to keep the control as is (from the previous sampling instant), or to switch to a backup controller.
- A more systematic approach is to relax the constraints in some way; this is often referred to as *constraint management*.

References

[1] J.B. Rawlings, D.Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory, Computation, and Design, 2nd edition*. Nob Hill Publishing 2017.
Available online at `https://sites.engineering.ucsb.edu/~jbraw/mpc`

[2] G. Goodwin, M.M. Seron, and J.A. De Don·. *Constrained Control and Estimation*. Springer 2004.
Available online via Chalmers Library.

[3] F Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*.
Available online at http://www.mpc.berkeley.edu/mpc-course-material

[4] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall 2002.

[5] S. Boyd and L. Vandenberghe. *Convex optimisation*. Cambridge University Press 2004.

[6] Diehl, M. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.

CHALMERS
UNIVERSITY OF TECHNOLOGY