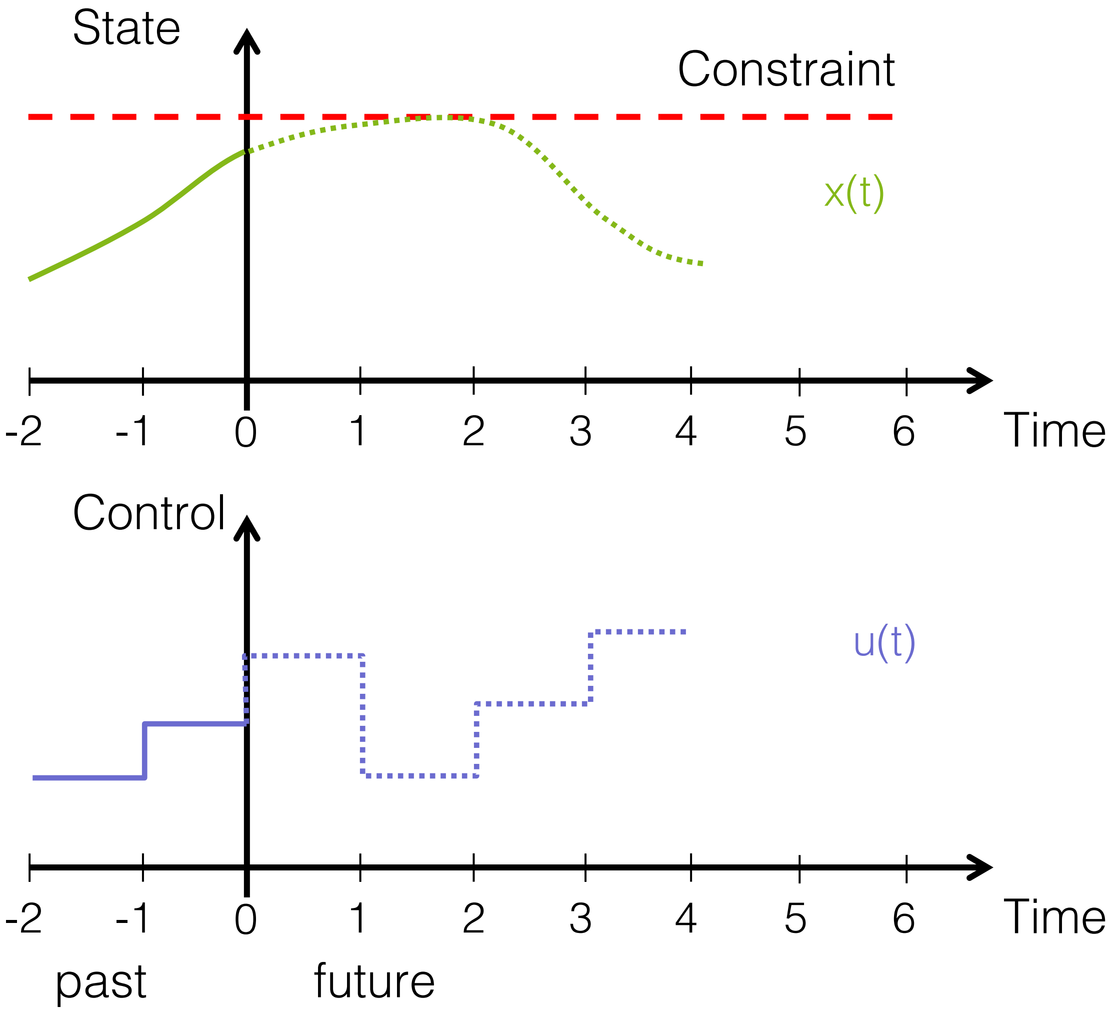
tu dortmund university

Fakultät Bio- und Chemieingenieurwesen

Multi-stage robust nonlinear model predictive control

Prof. Sergio Lucia
Chair of Process Automation Systems
Department of Biochemical and Chemical Engineering

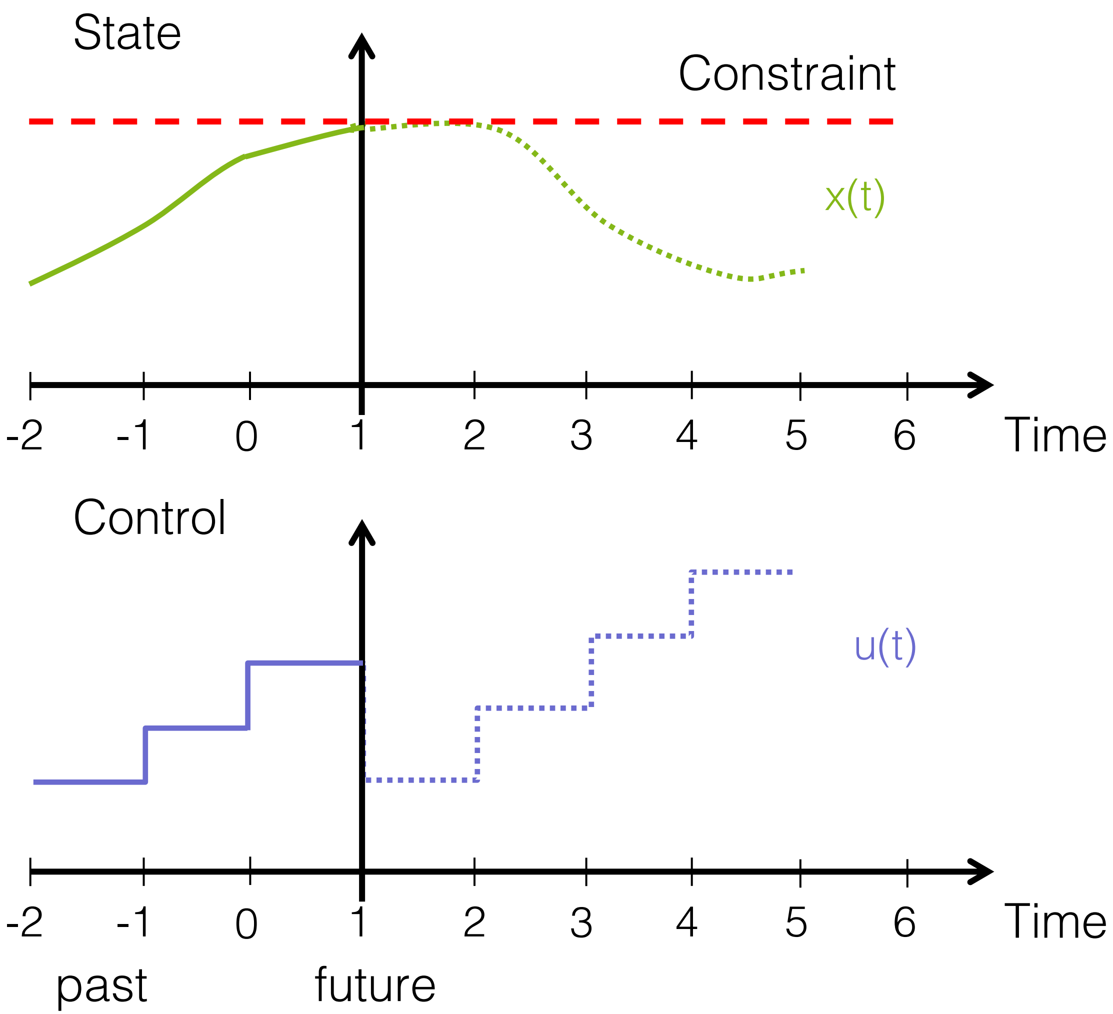
Model Predictive Control



Model Predictive Control

1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input

Model Predictive Control

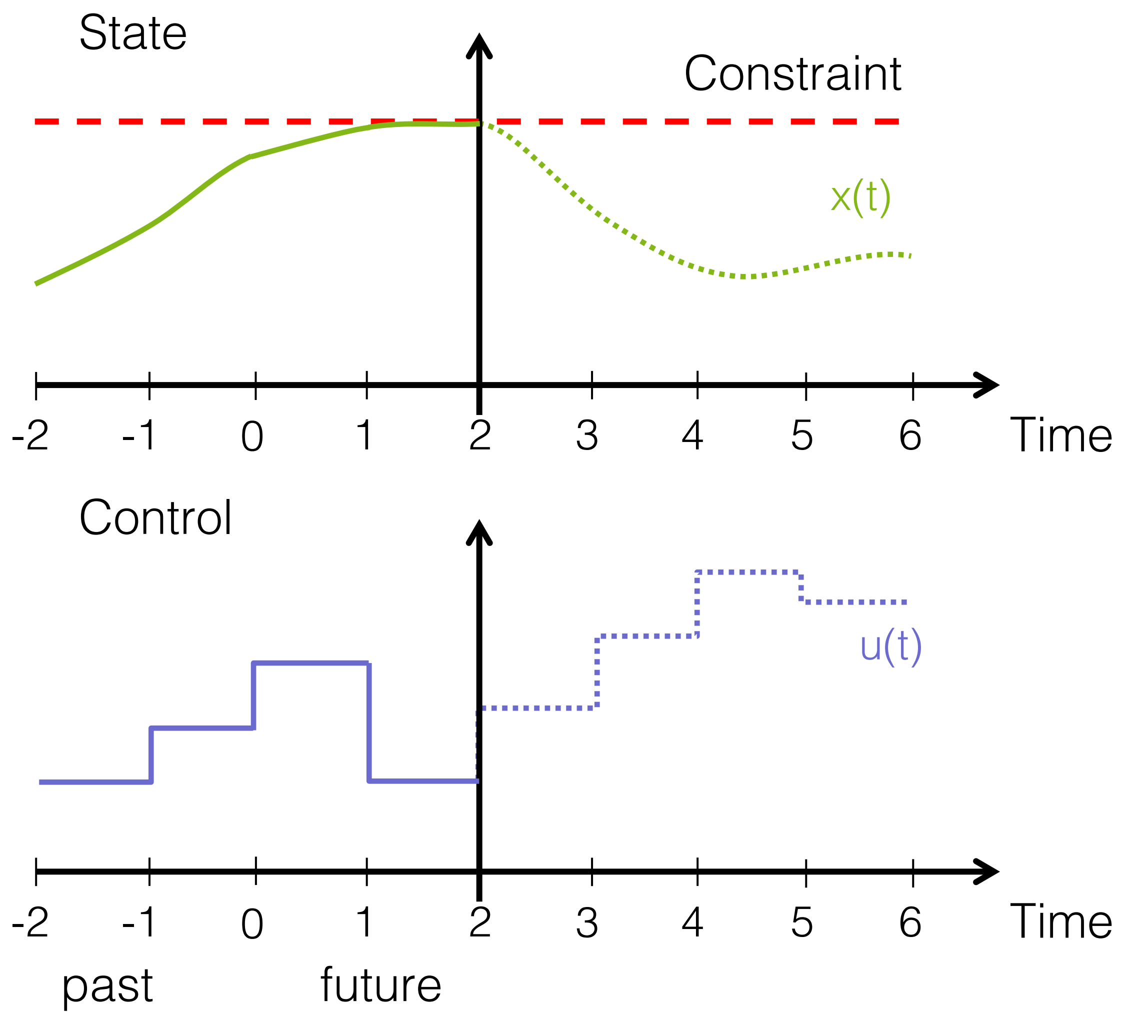


Model Predictive Control

1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input

-> Take new measurements and repeat

Model Predictive Control



Model Predictive Control

1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input

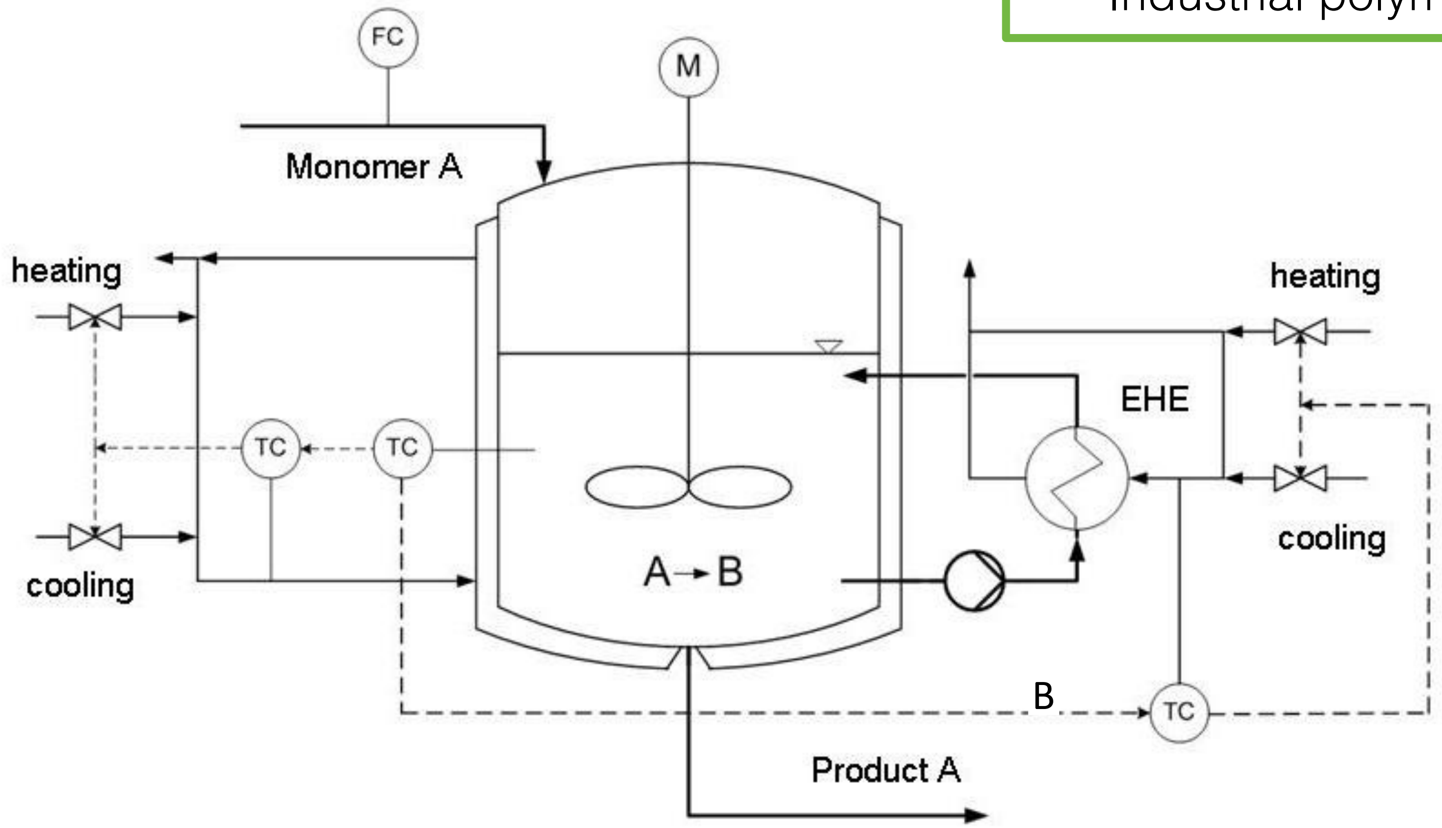
-> Take new measurements and repeat

Mathematical formulation

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, u_k), \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

An example of model predictive control

Industrial polymerization reactor



An industrial batch polymerization reactor

8 differential states
3 control inputs

$$\dot{m}_W = \dot{m}_{W,F}$$

$$\dot{m}_A = \dot{m}_{A,F} - k_{R1} m_{A,R} - \frac{p_1 k_{R2} m_{AWT} m_A}{m_{ges}}$$

$$\dot{m}_P = k_{R1} m_{A,R} + \frac{p_1 k_{R2} m_{AWT} m_A}{m_{ges}}$$

$$\dot{T}_R = \frac{1}{c_{p,R} m_{ges}} \left[\dot{m}_F c_{p,F} (T_F - T_R) + \Delta H_R k_{R1} m_{A,R} - k_K A (T_R - T_S) - \dot{m}_{AWT} c_{p,R} (T_R - T_{EK})\right]$$

$$\dot{T}_S = 1/(c_{p,S} m_S)[k_K A (T_R - T_S) - k_K A (T_S - T_M)]$$

$$\dot{T}_M = \frac{1}{c_{p,W} m_{M,KW}} \left[\dot{m}_{M,KW} c_{pW} (T_M^{IN} - T_M) + k_K A (T_S - T_M)\right]$$

$$\dot{T}_{EK} = \frac{1}{c_{p,R} m_{AWT}} \left[\dot{m}_{AWT} c_{p,W} (T_R - T_{EK}) - \alpha (T_{EK} - T_{AWT}) + \frac{p_1 k_{R2} m_A m_{AWT} \Delta \mathrm{H}_R}{m_{ges}}\right]$$

$$\dot{T}_{AWT} = \frac{1}{c_{p,W} m_{AWT,KW}} \left[\dot{m}_{AWT,KW} c_{p,W} (T_{AWT}^{IN} - T_{AWT}) - \alpha (T_{AWT} - T_{EK})\right]$$

$$k_{R1} = k_0 e^{-\frac{E_a}{R T_R}} (k_{U1}(1 - U) + k_{U2} U)$$

$$k_{R2} = k_0 e^{-\frac{E_a}{R T_{EK}}} (k_{U1}(1 - U) + k_{U2} U)$$

[Lucia, Andersson, Brand, Diehl and Engell, Journal of Process Control, 2014]

MPC with a tracking cost function

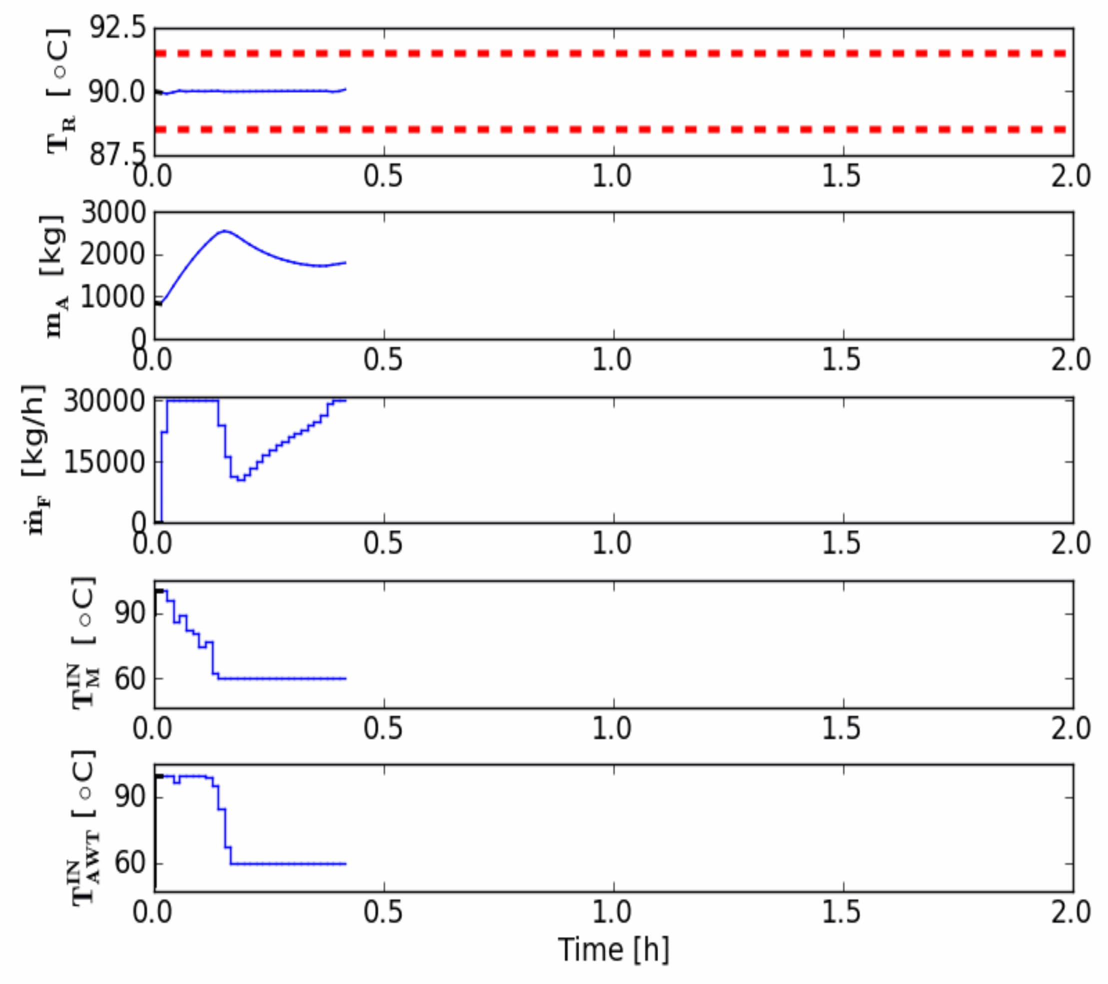
Reactor temperature

Monomer mass

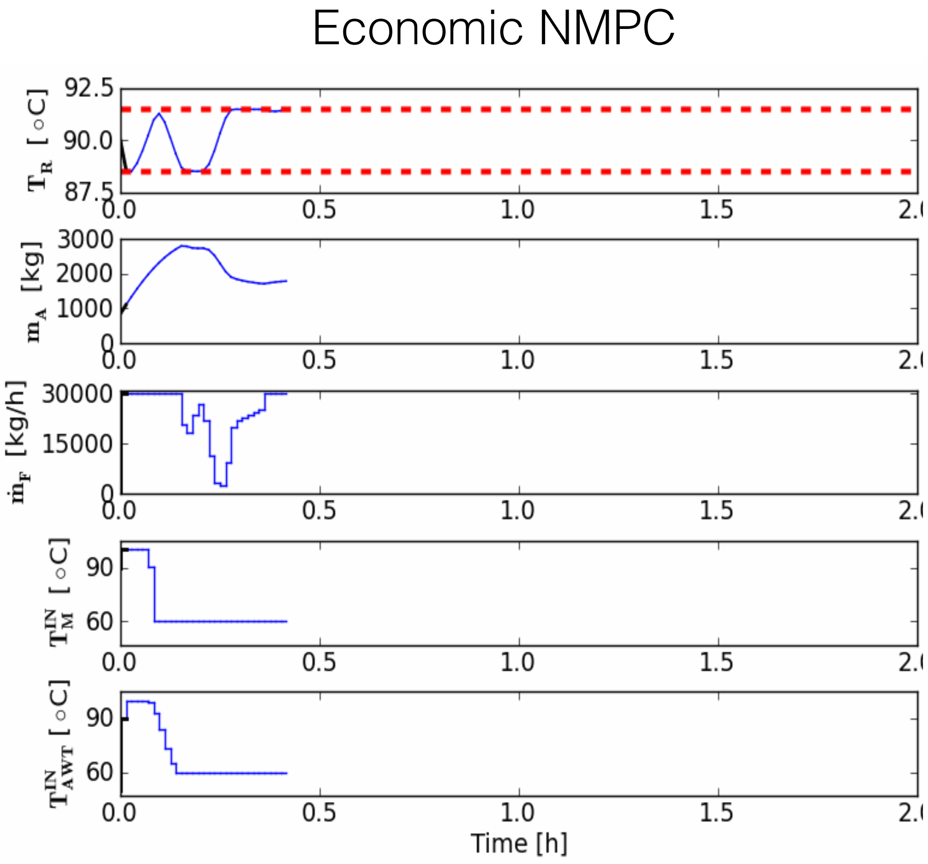
Monomer feed rate

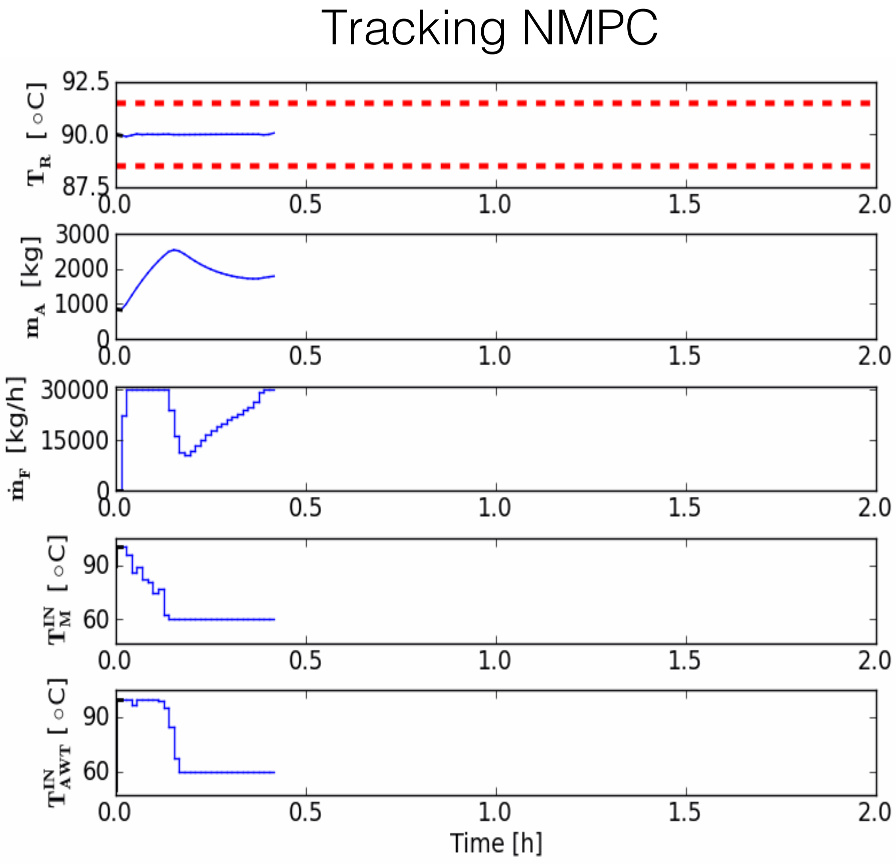
Jacket inlet temperature

EHE inlet temperature



Different cost functions: economic vs. tracking

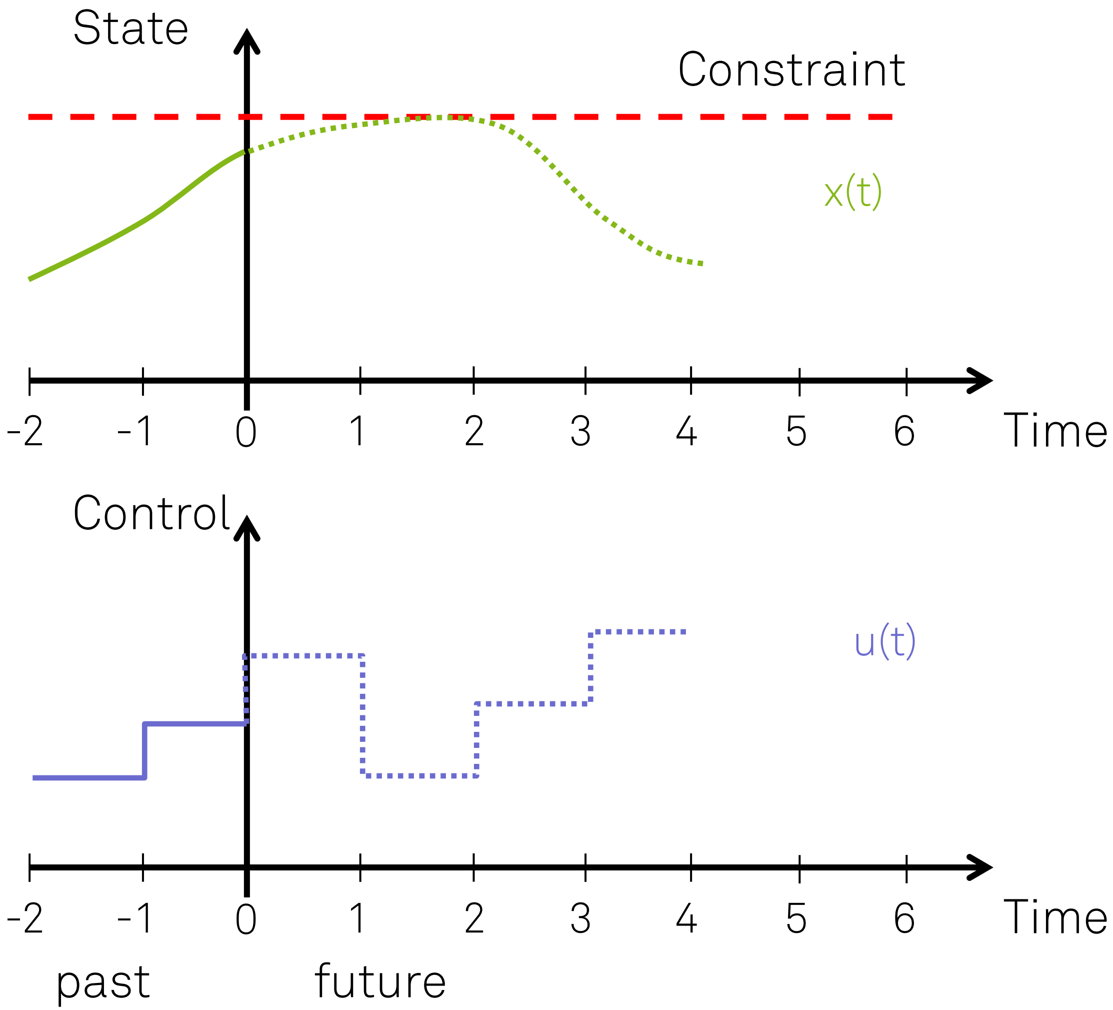




Maximize polymer production

The need for robust MPC

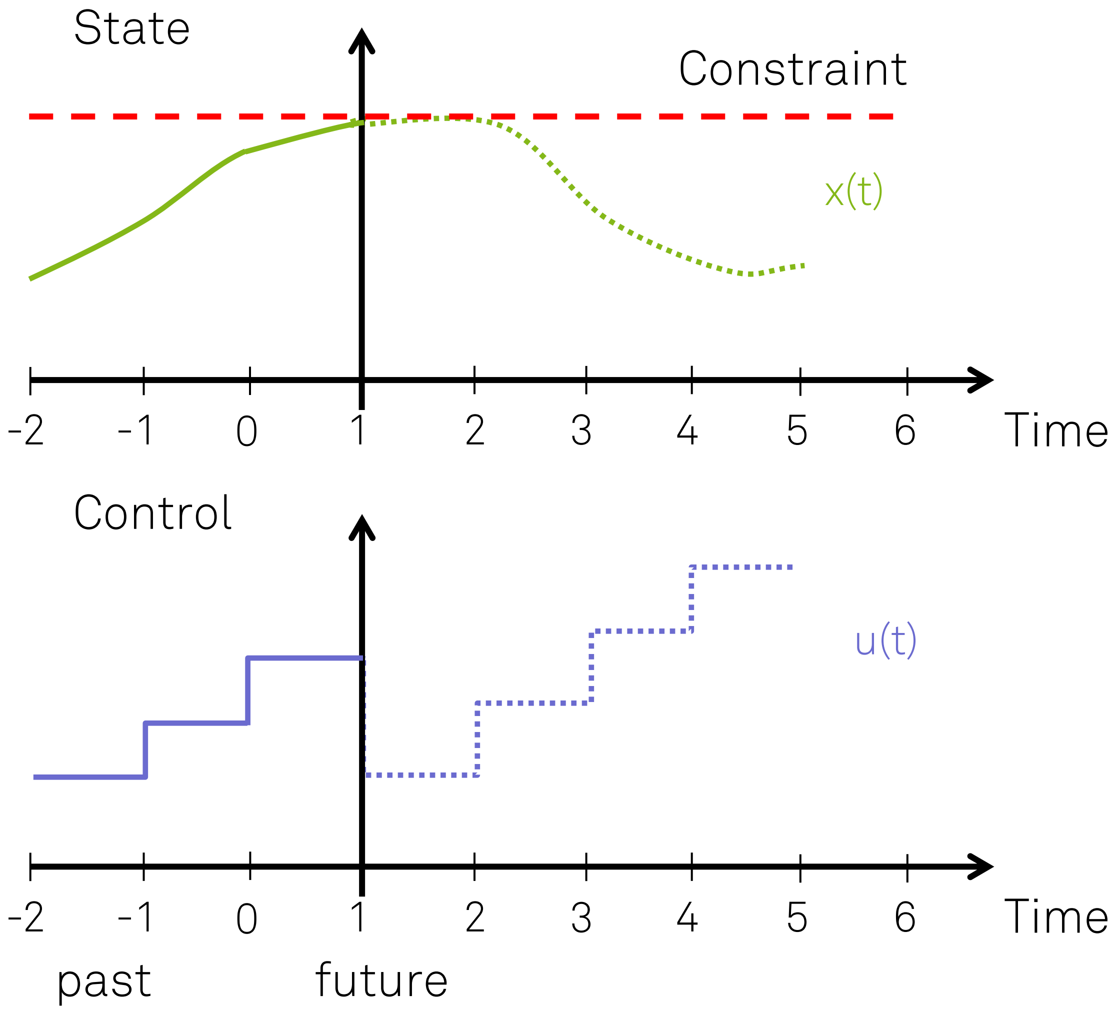
Model Predictive Control



Model Predictive Control

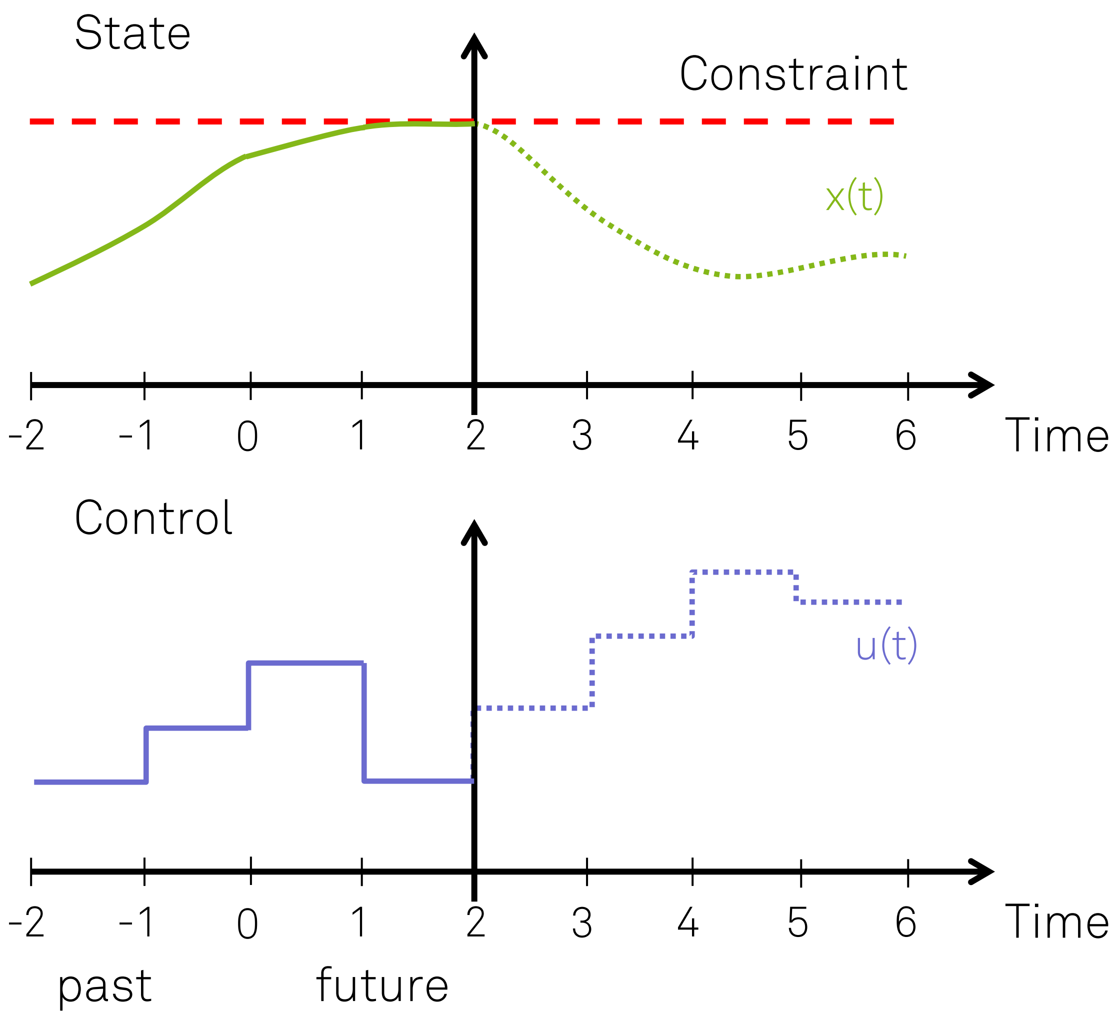
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input

Model Predictive Control



Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

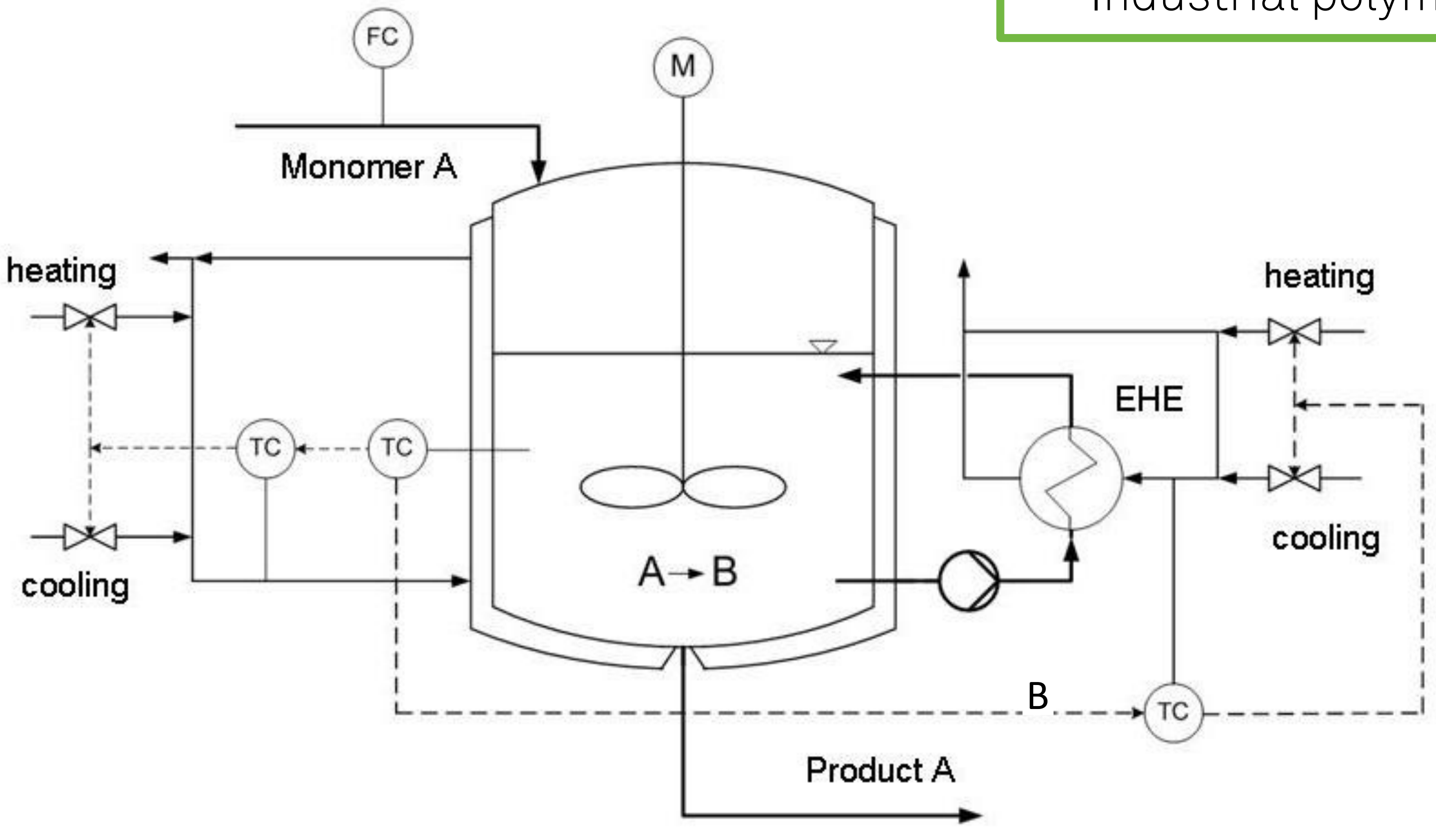
Model Predictive Control



Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

An example of model predictive control

Industrial polymerization reactor



An industrial batch polymerization reactor

8 differential states
3 control inputs
2 uncertain parameters

$$\dot{m}_W = \dot{m}_{W,F}$$

$$\dot{m}_A = \dot{m}_{A,F} - k_{R1} m_{A,R} - \frac{p_1 k_{R2} m_{AWT} m_A}{m_{ges}}$$

$$\dot{m}_P = k_{R1} m_{A,R} + \frac{p_1 k_{R2} m_{AWT} m_A}{m_{ges}}$$

$$\dot{T}_R = \frac{1}{c_{p,R} m_{ges}} \left[\dot{m}_F c_{p,F} (T_F - T_R) + \Delta H_R k_{R1} m_{A,R} - k_K A (T_R - T_S) - \dot{m}_{AWT} c_{p,R} (T_R - T_{EK}) \right]$$

$$\dot{T}_S = 1/(c_{p,S} m_S) [k_K A (T_R - T_S) - k_K A (T_S - T_M)]$$

$$\dot{T}_M = \frac{1}{c_{p,W} m_{M,KW}} \left[\dot{m}_{M,KW} c_{pW} (T_M^{IN} - T_M) + k_K A (T_S - T_M) \right]$$

$$\dot{T}_{EK} = \frac{1}{c_{p,R} m_{AWT}} \left[\dot{m}_{AWT} c_{p,W} (T_R - T_{EK}) - \alpha (T_{EK} - T_{AWT}) + \frac{p_1 k_{R2} m_A m_{AWT} \Delta \mathrm{H}_R}{m_{ges}} \right]$$

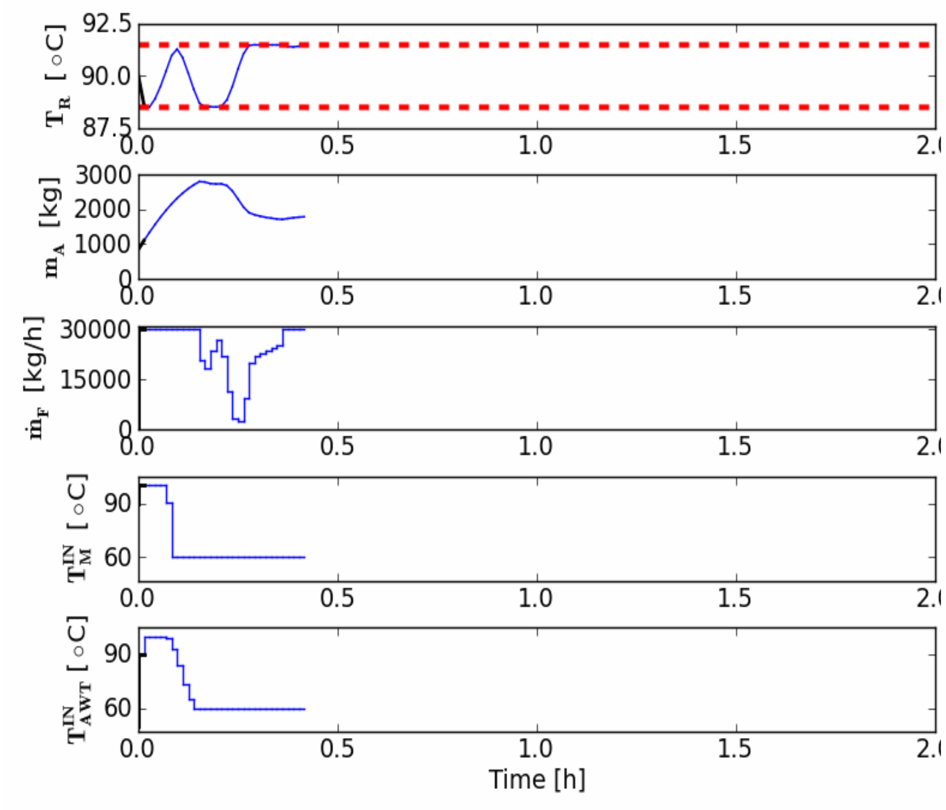
$$\dot{T}_{AWT} = \frac{1}{c_{p,W} m_{AWT,KW}} \left[\dot{m}_{AWT,KW} c_{p,W} (T_{AWT}^{IN} - T_{AWT}) - \alpha (T_{AWT} - T_{EK}) \right]$$

$$k_{R1} = k_0 e^{-\frac{E_a}{R T_R}} (k_{U1}(1 - U) + k_{U2} U)$$

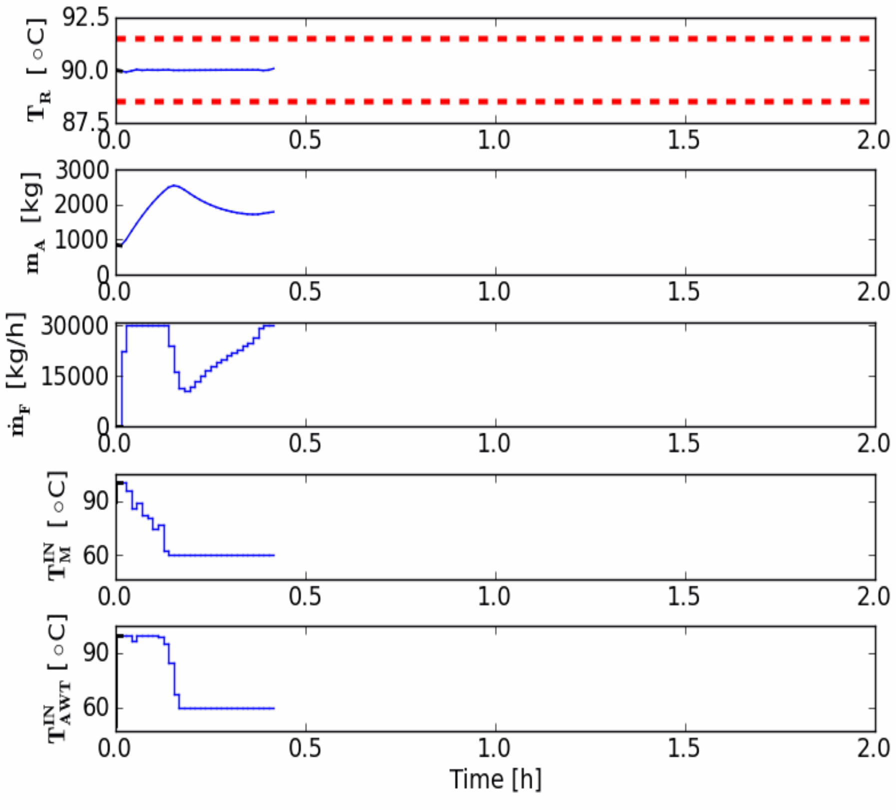
$$k_{R2} = k_0 e^{-\frac{E_a}{R T_{EK}}} (k_{U1}(1 - U) + k_{U2} U)$$

Different cost functions: economic vs. tracking

Economic NMPC

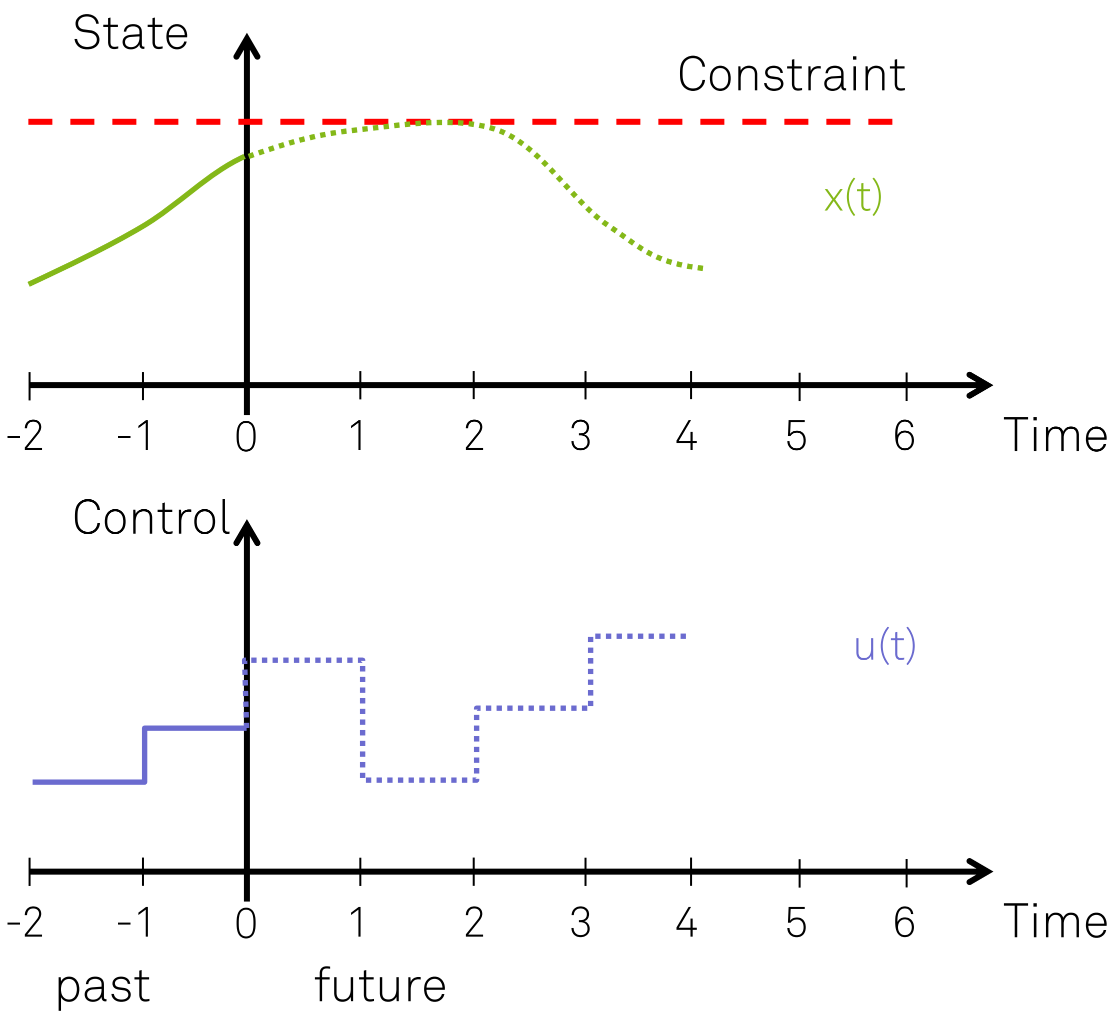


Tracking NMPC



Maximize polymer production

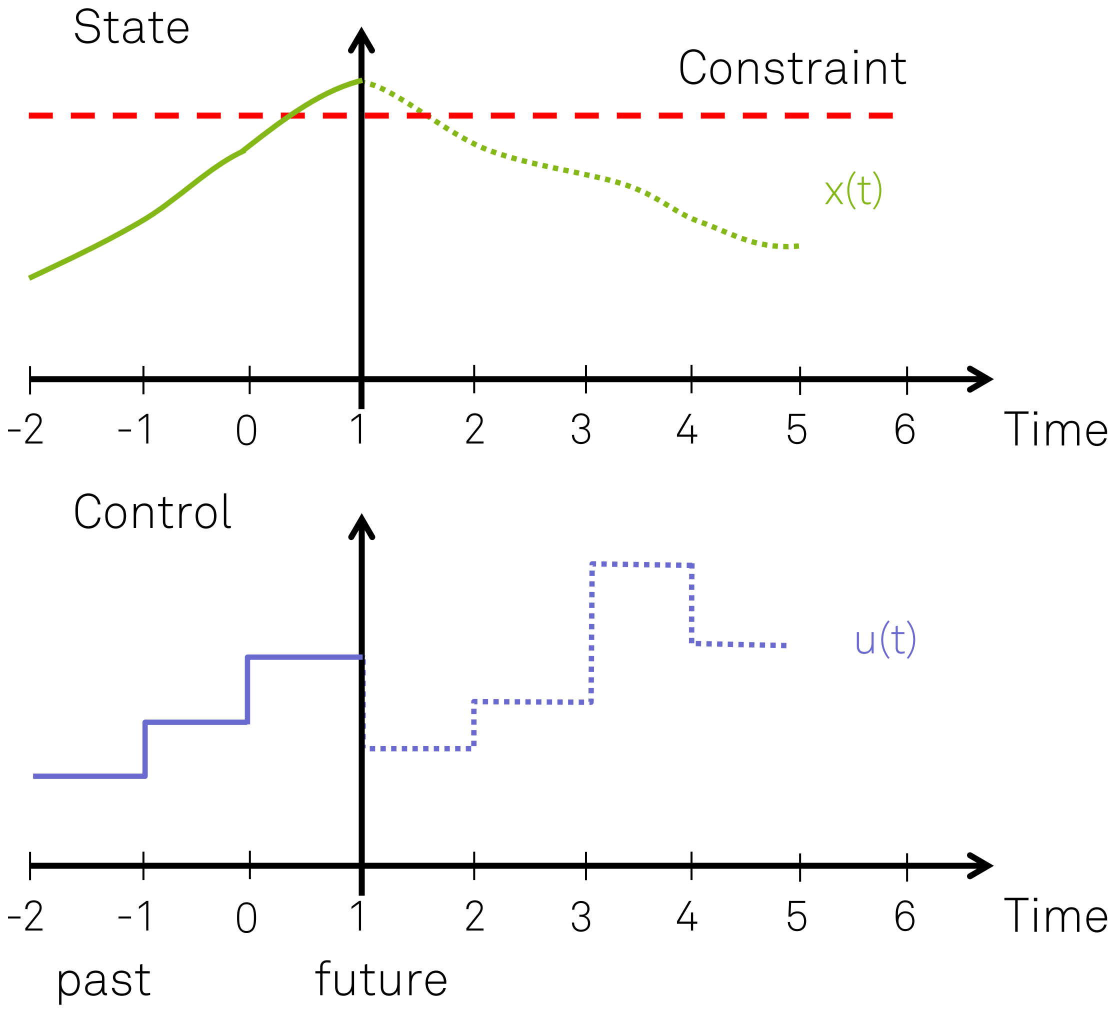
Model predictive control with a wrong model



Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

What if the model is not exact?

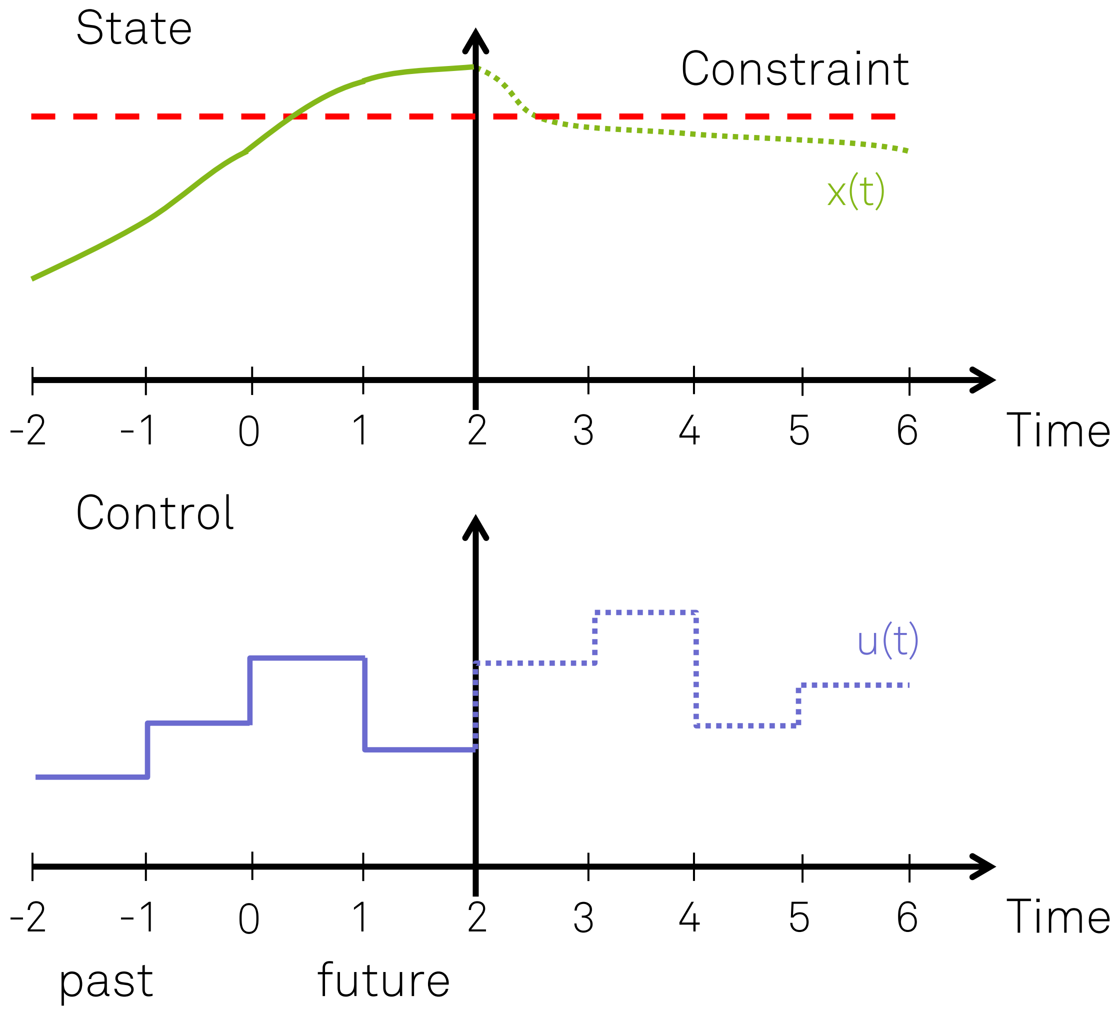
Model predictive control with a wrong model



Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

What if the model is not exact?

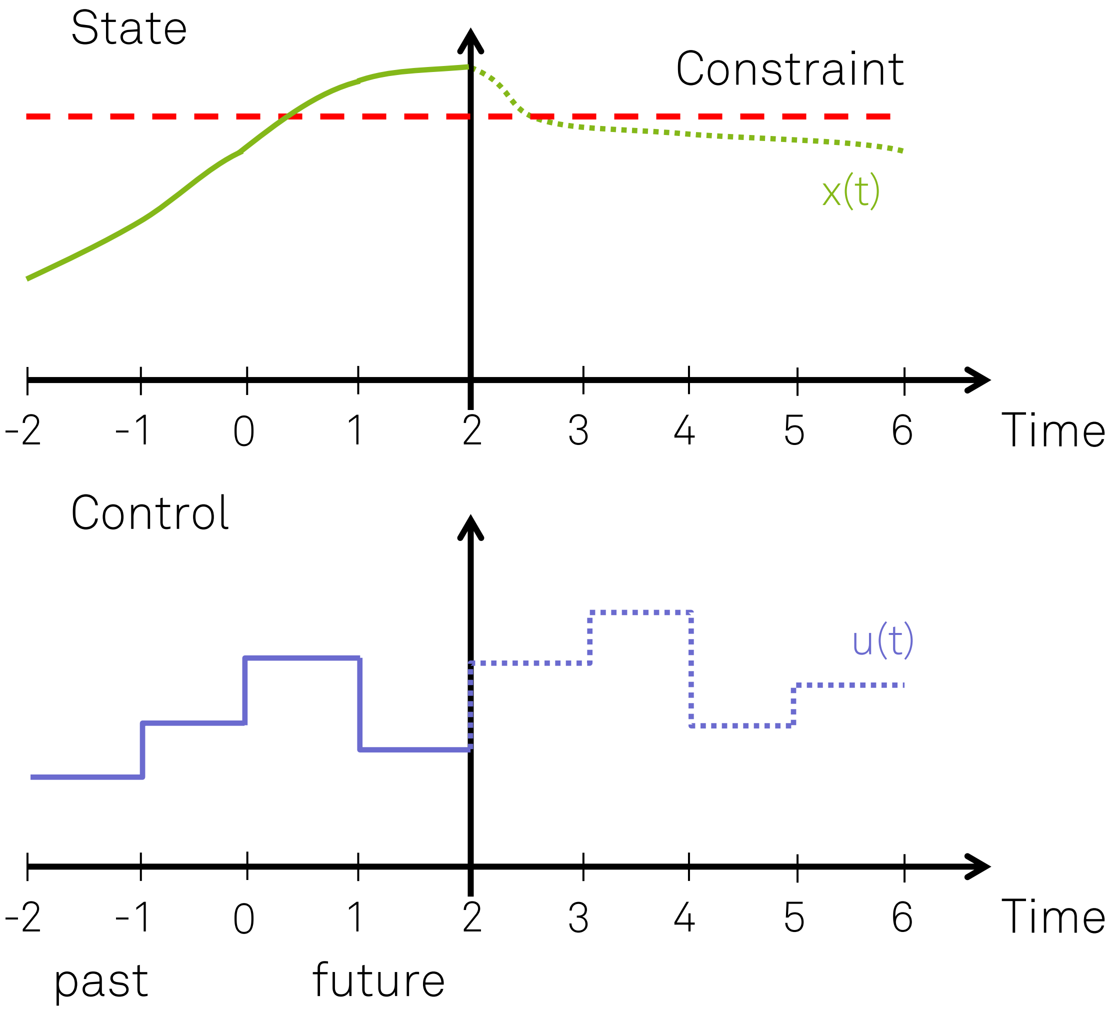
Model predictive control with a wrong model



Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

What if the model is not exact?
- Violation of constraints
- Decreased performance
- Instability

Model predictive control with a wrong model



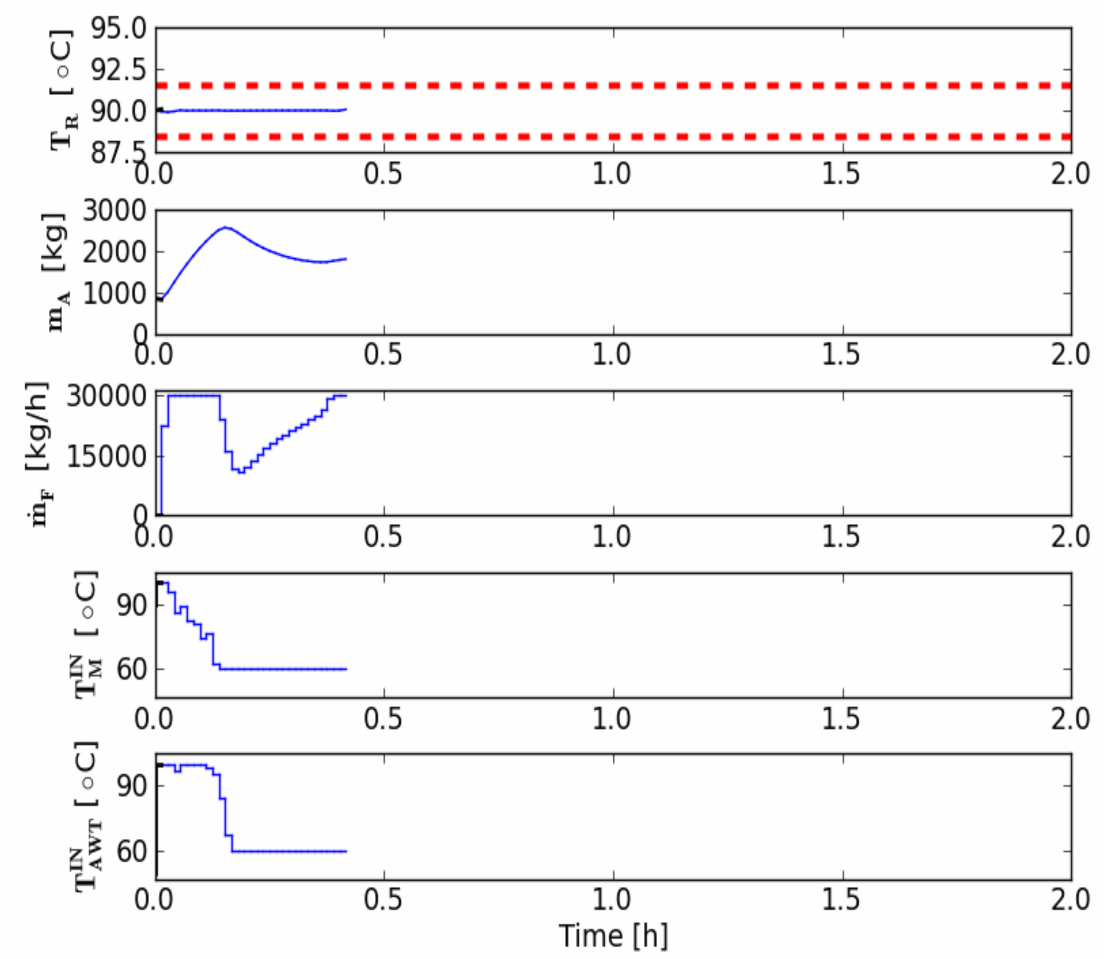
Model Predictive Control
1. Measure / Estimate state
2. Solve optimization problem
   - Mathematical model
   - Cost function
   - Constraints
3. Apply first control input
-> Take new measurements and repeat

What if the model is not exact?
- Violation of constraints
- Decreased performance
- Instability

Robust MPC

Model predictive control with a wrong model

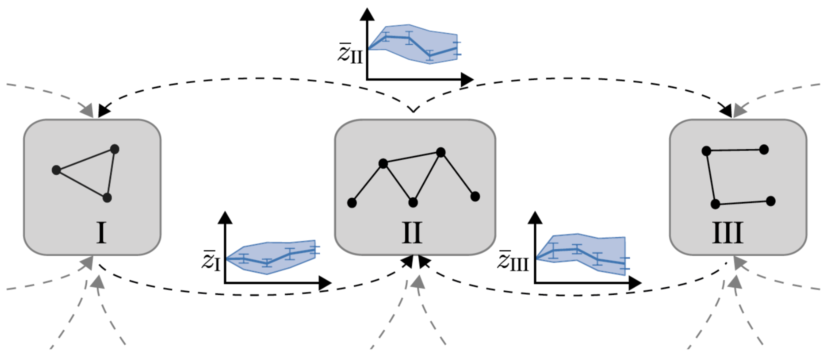
Standard NMPC: uncertainty + 15%



Sources of uncertainty are everywhere

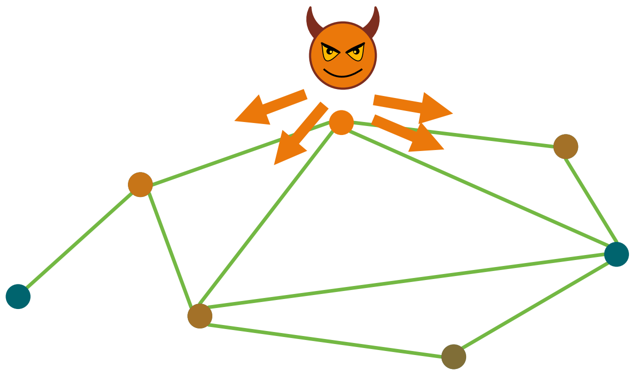
Uncertainty also appears in complex systems because of...

complexity encapsulation



[Lucia, Kögel and Findeisen, Annual reviews in Control, 2016]

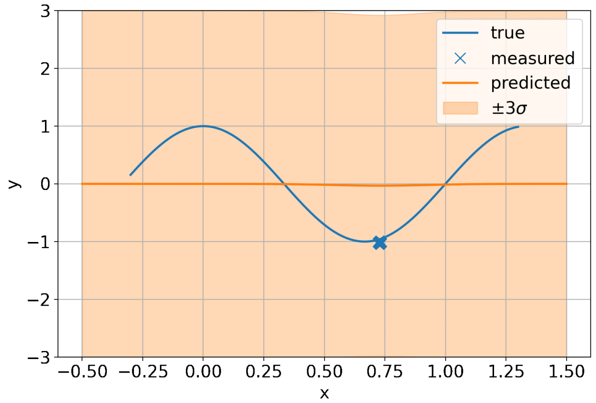
faults / malicious attacks



[Braun, Albrecht, Lucia, IEEE Tran. Aut. Control, 2022]

[Braun, Albrecht, Lucia, ICINCO Conference, Porto, 2022]

data-based models



[Fiedler and Lucia, IEEE Access 2023]

[Johnson Fiedler and Lucia, ADCHEM 2024]

Challenges of robust MPC

Objective, constraints, performance and stability

Robust MPC: Modeling uncertainty

Some methods differentiate between **two types of uncertainty**

- Additive disturbances

$$x_{k+1} = f(x_k, u_k) + w_k, \quad w_k \in \mathbb{W}$$

- Parametric uncertainty

$$x_{k+1} = f(x_k, u_k, d_k), \quad d_k \in \mathbb{D}$$

Robust MPC: Mathematical formulation

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, d_k) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k, d_k), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, u_k, d_k), \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

- How to deal with the uncertainty in cost and constraints?

Robust MPC: Four main challenges

1. Which objective should be optimized?

2. Constraints should be satisfied for all possible outcomes

3. Performance should not be overly conservative

4. The closed-loop should be stable for all possible outcomes

1. Choice of objective function

It affects the performance and stability of the closed-loop

Several possibilities:

- Consider the nominal cost

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, 0) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k, 0), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, u_k, d_k), \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

1. Choice of objective function

It affects the performance and stability of the closed-loop

Several possibilities:

- Consider the nominal cost
- Consider a min-max (worst-case) cost

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}}\ \underset{\mathbf{d}}{\text{maximize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, d_k) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k, d_k), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, u_k, d_k), \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

2. Constraints satisfied for all possible outcomes

Usually named **robust constraint satisfaction**

Problem: infinitely many constraints

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, d_k) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k, d_k), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, u_k, d_k), \forall d_k \in \mathbb{D}, \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

2. Constraints satisfied for all possible outcomes

Usually named **robust constraint satisfaction**

Problem: infinitely many constraints

- Reformulate to obtain finitely many constraints

$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k, 0) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, u_k, d_k), x_0 = x_{\text{init}} \\
& 0 \geq \underset{\mathbf{d}}{\text{maximize}}\ g(x_k, u_k, d_k), \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}
$$

3. Performance should not be overly conservative

Traditional approach: *open-loop robust MPC*

- Minimize a cost function for the worst-case uncertainty value
- The same sequence of control inputs to satisfy the constraints

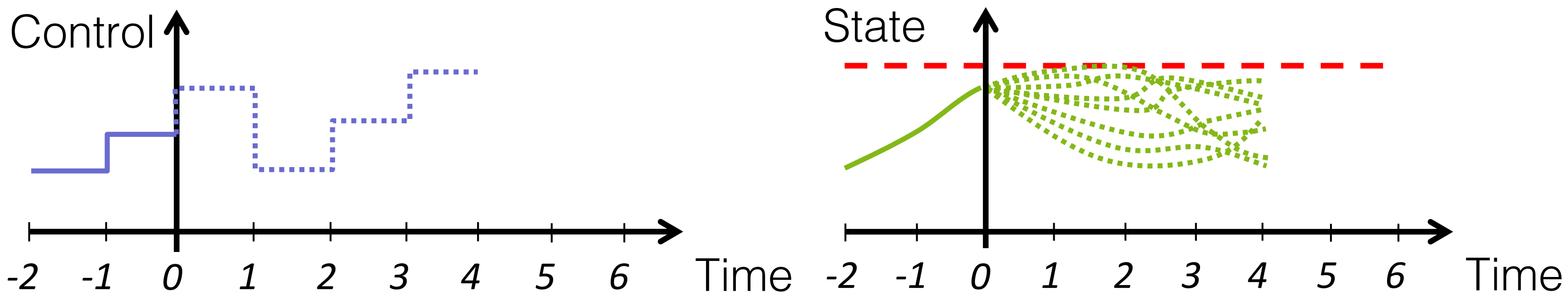
$$\boldsymbol{u} = \{u(0), \ldots, u(N-1)\}$$

3. Performance should not be overly conservative

Traditional approach: *open-loop robust MPC*

- Minimize a cost function for the worst-case uncertainty value
- The same sequence of control inputs to satisfy the constraints

$$\boldsymbol{u} = \{u(0), \ldots, u(N-1)\}$$

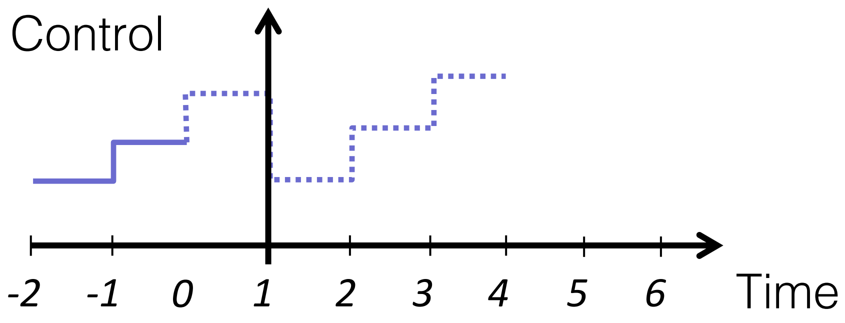


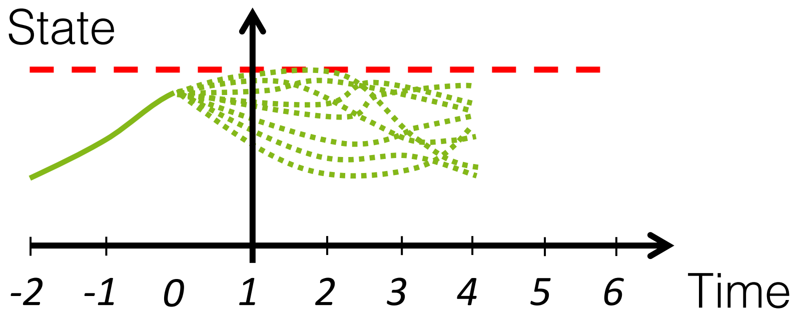
3. Performance should not be overly conservative

Traditional approach: *open-loop robust MPC*

- Minimize a cost function for the worst-case uncertainty value
- The same sequence of control inputs to satisfy the constraints

$$\boldsymbol{u} = \{u(0), \dots, u(N-1)\}$$



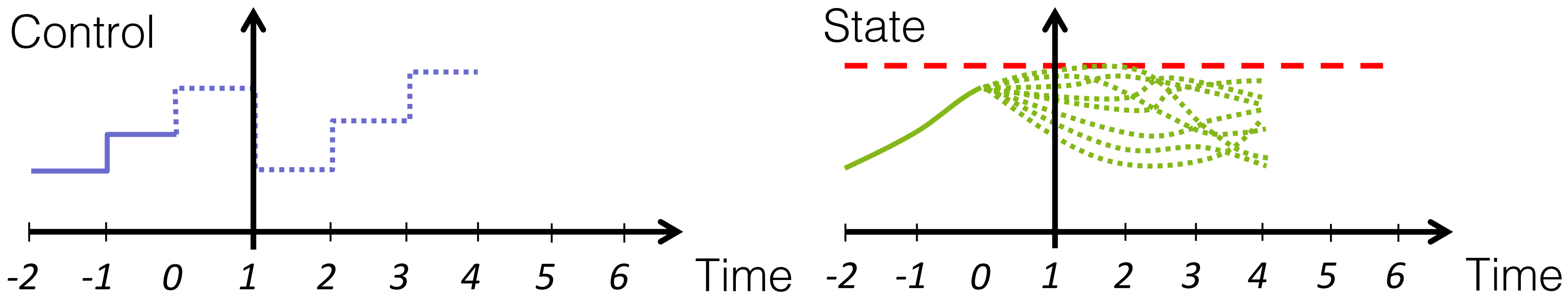


3. Performance should not be overly conservative

Traditional approach: *open-loop robust MPC*

- Minimize a cost function for the worst-case uncertainty value
- The same sequence of control inputs to satisfy the constraints

$$\boldsymbol{u} = \{u(0), \ldots, u(N-1)\}$$



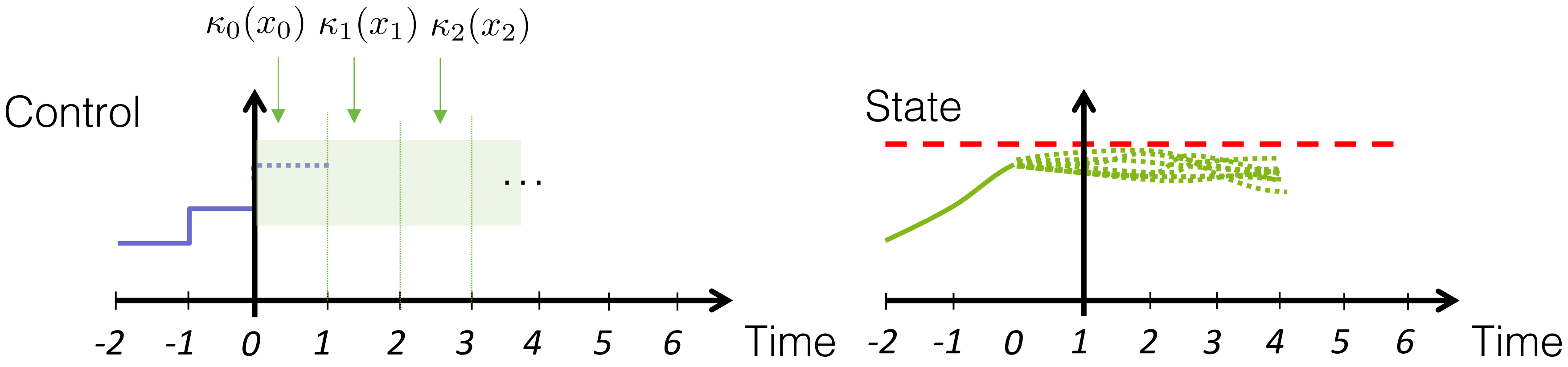
(Can be) Very conservative!

3. Performance should not be overly conservative

Closed-loop (or feedback) robust MPC

- Minimize a cost function for the worst-case uncertainty value over a sequence of **control policies**

$$\boldsymbol{\kappa} = \{\kappa_0(\cdot), \dots, \kappa_{N-1}(\cdot)\}$$



3. Performance should not be overly conservative

Compute a sequence of optimal control **policies** $\boldsymbol{\kappa} = \{\kappa_0(\cdot), \ldots, \kappa_{N-1}(\cdot)\}$

$$\begin{aligned}
\underset{\boldsymbol{\kappa},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell(x_k, \kappa_k(x_k), d_k) + V_f(x_N) \\
\text{subject to:} \quad & x_{k+1} = f(x_k, \kappa_k(x_k), d_k), x_0 = x_{\text{init}} \\
& 0 \geq g(x_k, \kappa_k(x_k), d_k), \forall d_k \in \mathbb{D}, \\
& 0 \geq g_f(x_N), \\
& k \in [0, N-1]
\end{aligned}$$

3. Performance should not be overly conservative

Computing general policies $\boldsymbol{\kappa} = \{\kappa_0(\cdot), \dots, \kappa_{N-1}(\cdot)\}$ is challenging:

- Infinite dimensional optimization problem

Usually **restrict the structure of the policy**

- Use state-affine control policies $\mu_k(x) = Kx + v_k$
- Use uncertainty-affine control policies $\mu_k(w_k) = Kw_k + v_k$

[Goulart, Kerrigan and Maciejowski, Automatica, 2006]

Suboptimal but tractable optimization problems

Multi-stage MPC

A closed-loop approach with a scenario tree representation of the uncertainty

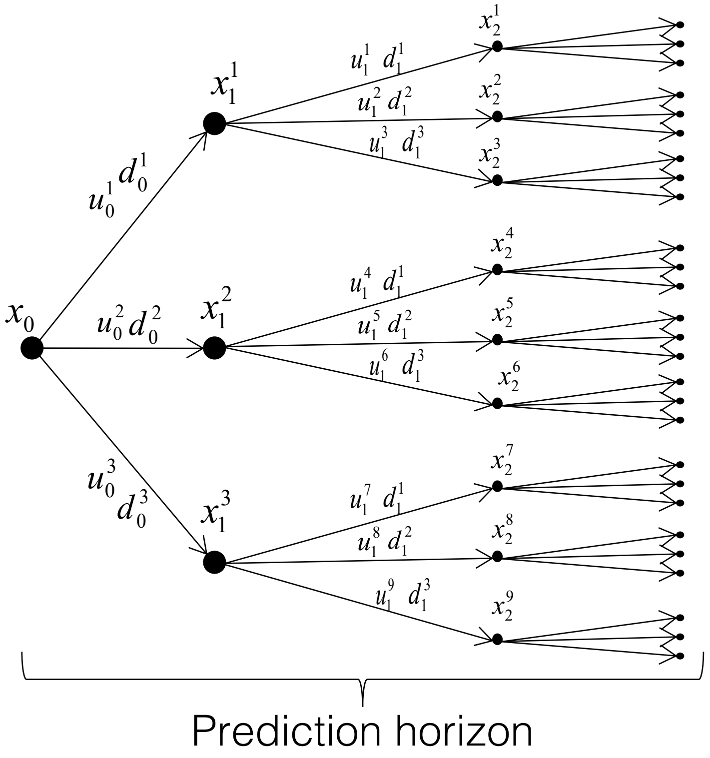
Multi-stage stochastic programming

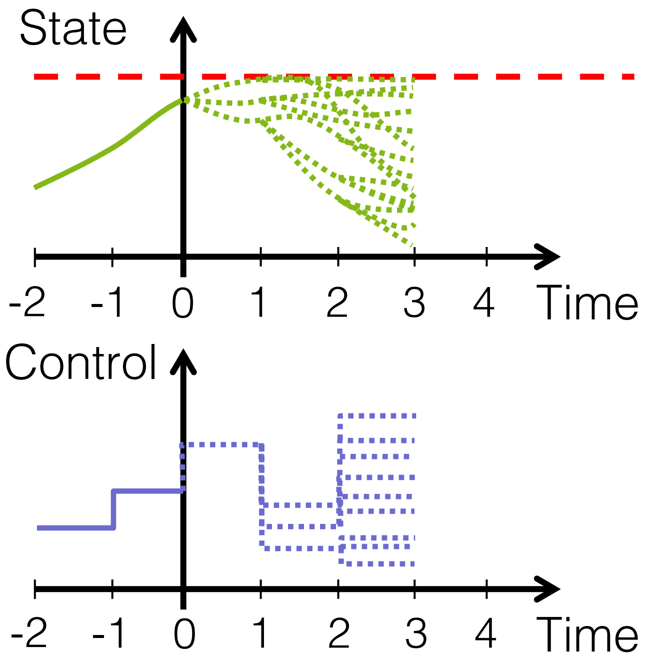
- Multi-stage stochastic programming is an extension of linear programming to deal with uncertainty
  [Dantzig & Infanger, 1993]

- Modeling of sequential decision-making problems via scenario trees

- In context of MPC, first used to model minmax MPC problems
  [Scokaert and Mayne, TAC 1998]

Multi-stage NMPC: Formulation



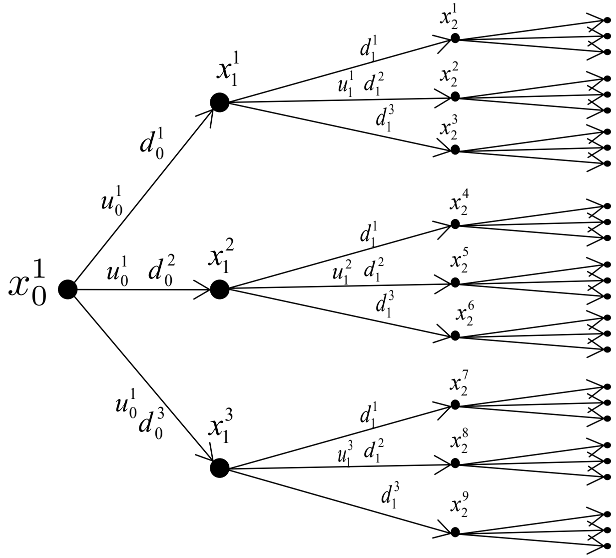


Decisions with the same information must be equal

- Non anticipativity constraints

[Scokaert and Mayne, TAC 1998], [Diehl, TAC, 2007]
[Bernardini and Bemporad, CDC, 2009], [De la Peña, Bemporad, Alamo, CDC 2005]
[Lucia, Finkler and Engell, Journal of Process Control, 2013][Lucia, Andersson, Brand, Diehl and Engell, Journal of Process Control, 2014]
[Krishnamoorthy & Skogestad, 2018, 2019], [Yang & Biegler, JPC, 2013], [and many others]

Mathematical formulation



$$
\begin{aligned}
\underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \sum_{i \in \mathcal{I}_{\text{tree}}(k)} \omega_k^i \ell(x_k^i, u_k^i, d_k^{r(i)}) + \sum_{i \in \mathcal{I}_{\text{tree}}(N)} \omega_N^i V_f(x_N^i) \\
\text{subject to:} \quad x_{k+1}^{\text{child}(i,k)} &= f(x_k^i, u_k^i, d_k^{r(i)}), x_0 = x_{\text{init}} \\
0 &\geq g(x_k^i, u_k^i, d_k^{r(i)}), \\
0 &\geq g_f(x_N^i), \\
(i,k) &\in \mathcal{I}_{\text{tree}}
\end{aligned}
$$

The importance of a closed-loop formulation

An illustrative example [modified from Scokaert & Mayne, TAC 1998]

$$x_{k+1} = x_k + u_k + d_k \qquad -1.5 \leq x_k \leq 1.5$$

$$d_k \in [-1, 1] \qquad -1.5 \leq u_k \leq 1.5$$

$$N = 3$$

The importance of a closed-loop formulation

An illustrative example [modified from Scokaert & Mayne, TAC 1998]

$$x_{k+1} = x_k + u_k + d_k \qquad -1.5 \leq x_k \leq 1.5$$

$$d_k \in [-1, 1] \qquad -1.5 \leq u_k \leq 1.5$$

$$N = 3$$

Consider an **open-loop formulation:**

- If $d_0 = d_1 = d_2 = 1$ then:

$$x_3 = x_0 + u_0 + u_1 + u_2 + 1 + 1 + 1 = x_0 + u_0 + u_1 + u_2 + 3$$

- If $d_0 = d_1 = d_2 = -1$ then:

$$x_3 = x_0 + u_0 + u_1 + u_2 - 1 - 1 - 1 = x_0 + u_0 + u_1 + u_2 - 3$$

The importance of a closed-loop formulation

An illustrative example [modified from Scokaert & Mayne, TAC 1998]

$$x_{k+1} = x_k + u_k + d_k \qquad -1.5 \leq x_k \leq 1.5$$

$$d_k \in [-1, 1] \qquad -1.5 \leq u_k \leq 1.5$$

$$N = 3$$

Consider an **open-loop formulation:**

- If $d_0 = d_1 = d_2 = 1$ then:

$$x_3 = x_0 + u_0 + u_1 + u_2 + 1 + 1 + 1 = x_0 + u_0 + u_1 + u_2 + 3$$

- If $d_0 = d_1 = d_2 = -1$ then:

$$x_3 = x_0 + u_0 + u_1 + u_2 - 1 - 1 - 1 = x_0 + u_0 + u_1 + u_2 - 3$$

If $x_0 = 0$, can you find **a common feasible solution** for all scenarios?

The importance of a closed-loop formulation

For scenario 1

$$x_3^1 = x_0 + u_0^1 + u_1^1 + u_2^1 + 3$$

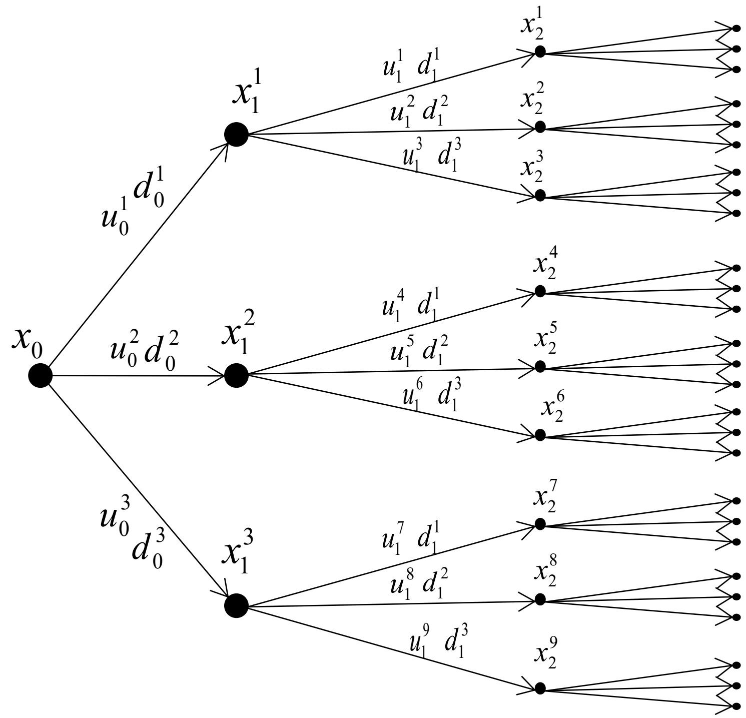
For scenario 27

$$x_3^{27} = x_0 + u_0^3 + u_1^9 + u_2^{27} - 3$$

Non-anticipativity constraints

$$u_0^1 = u_0^3$$

If $x_0 = 0$, can you find a feasible solution?



Formulation as an optimization problem

How to generate a scenario tree?

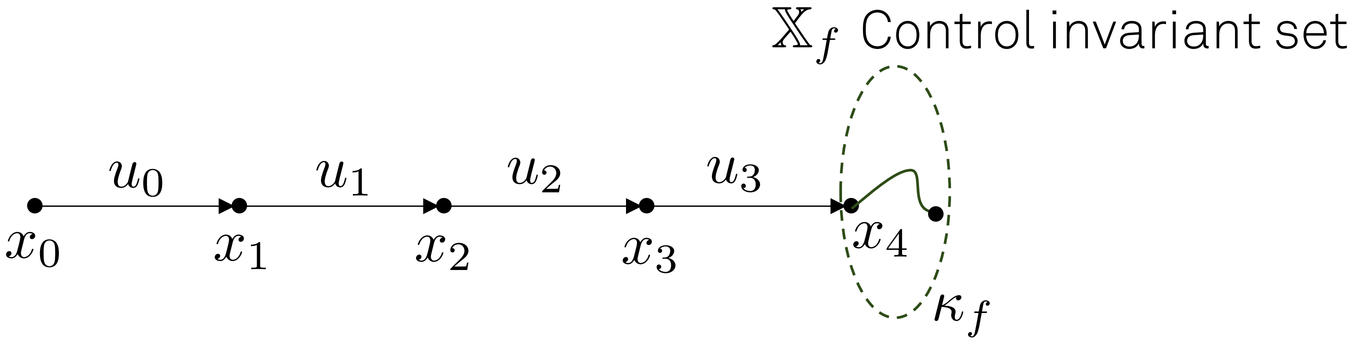
- Usually consider combinations of the extreme values $p_1 = [0.8, 1.2], p_2 = [45, 55]$
- Example: 2 uncertainties lead to four scenarios

$$d_1 = [0.8, 45] \quad d_3 = [1.2, 45]$$
$$d_2 = [0.8, 55] \quad d_4 = [1.2, 55]$$

- In the linear case, this guarantees robust constraint satisfaction (possible for polytopic uncertainty)
- In the general case, this is heuristic that very often works very well

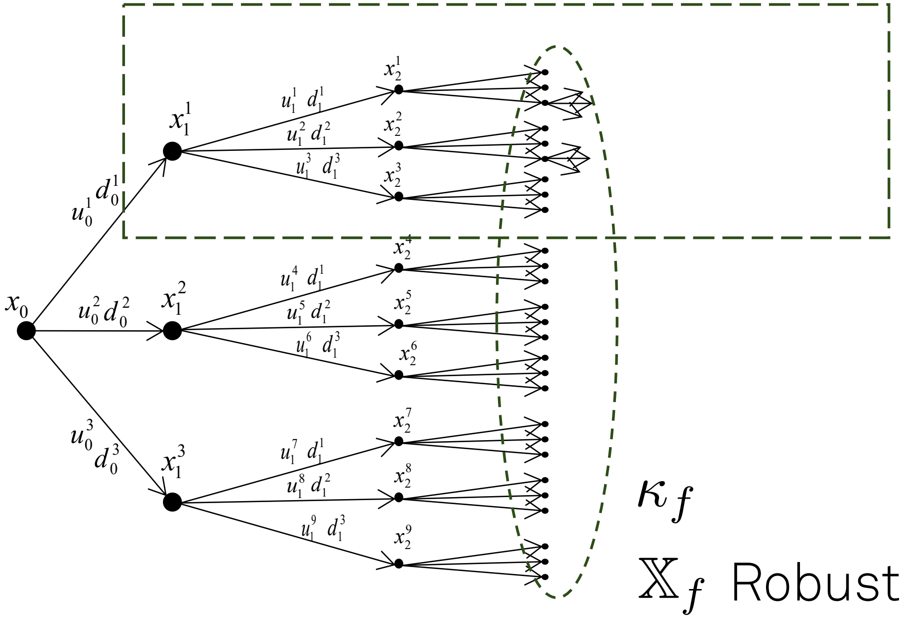
Theoretical properties of multi-stage MPC

- Need stage cost, terminal cost, terminal set and terminal control law that satisfy usual assumptions
- The weights of the scenarios are chosen *properly*
- You can establish input-to-state (practical) stability



Theoretical properties of multi-stage MPC

- Need stage cost, terminal cost, terminal set and terminal control law that satisfy usual assumptions
- The weights of the scenarios are chosen *properly*
- You can establish Input-to-state (practical) stability



κ_f

$\mathbb{X}_f$ Robust control invariant set

(iii) The main part of the stability analysis of MPC schemes is the descent property (6c). We write the value function of the problem with horizon N using Bellman's optimality principle as:

$$V_N(x_0^1) = \sum_{r=1}^{s} \pi^r V_{N-1}(f(x_0^1, u_0^1, d^r)) + \ell(x_0^1, u_0^1)$$
$$= \ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r))$$
$$+ \sum_{r=1}^{s} \pi^r V_{N-1}(f(x_0^1, u_0^1, d^r)) - \sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r))$$

Grouping summations and reordering we obtain:

$$- V_N(x_0^1) + \ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r))$$
$$= \sum_{r=1}^{s} \pi^r [V_N(f(x_0^1, u_0^1, d^r)) - V_{N-1}(f(x_0^1, u_0^1, d^r))]$$
$$\leq \sum_{r=1}^{s} \pi^r c_3 = c_4.$$

The inequality is true because of the boundedness property derived in Lemma 4. Then:

$$\ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r)) \leq V_N(x_0^1) + c_4, \qquad (16)$$

which implies

$$\sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r)) - V_N(x_0^1) \leq -\gamma_1(|x_0^1|) + c_4, \qquad (17)$$

for all $x_0^1 \in \mathbb{X}_A(N)$ and for all $d^r \in \mathbb{D}$. The inequality in (17) shows a weighted descent in the value function over all the possible realizations of the uncertainty. In order to prove ISpS of the closed loop system, a decrease for any realization is needed according to (6c). We exploit the fact that any $\mathcal{K}$-function ξ and finite positive real numbers a_1, a_2 fulfill the following inequality: $\xi(a_1 + a_2) \leq \xi(2a_1) + \xi(2a_2)$ as stated in [4,23]. Using this inequality and Lemma 5 it follows that for any realization of the uncertainty $d \in \mathbb{D}$:

$$V_N(f(x_0^1, u_0^1, d)) \leq \sum_{r=1}^{s} \pi^r V_N(f(x_0^1, u_0^1, d^r)) + \gamma_{av}(|d|) + c_5, \qquad (18)$$

with $\gamma_{av}(|d|) = \gamma_V \circ \gamma_{f_d}(2|d|)$ and $c_5 = \sum_{r=1}^{s} \pi^r \gamma_V \circ \gamma_{f_d}(2|d^r|) + \rho$. Then for all $x_0^1 \in \mathbb{X}_A(N)$ we have that:

$$V_N(f(x_0^1, u_0^1, d^r)) - V_N(x_0^1) \leq -\gamma_1(|x_0^1|) + \gamma_{av}(|d|) + c_4 + c_5 \qquad (19)$$

[Lucia, Limon and Engell, Systems and Control Letters, 2020]

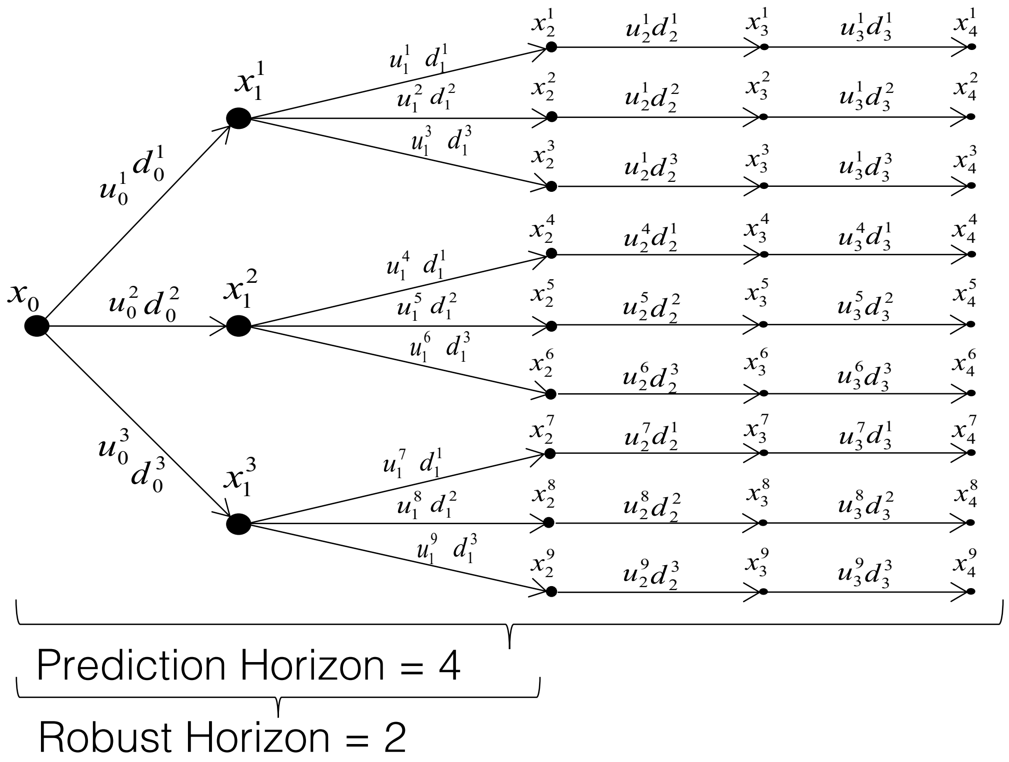
Multi-stage NMPC: Four main challenges

1. Which objective should be optimized?
   Weighted sum of scenarios
2. Constraints should be satisfied for all possible outcomes
   Constraints on all nodes of the tree
3. Performance should not be overly conservative
   Feedback via tree structure
4. The closed-loop should be stable for all possible outcomes
   IS(p)S is possible

Multi-stage NMPC: Robust horizon

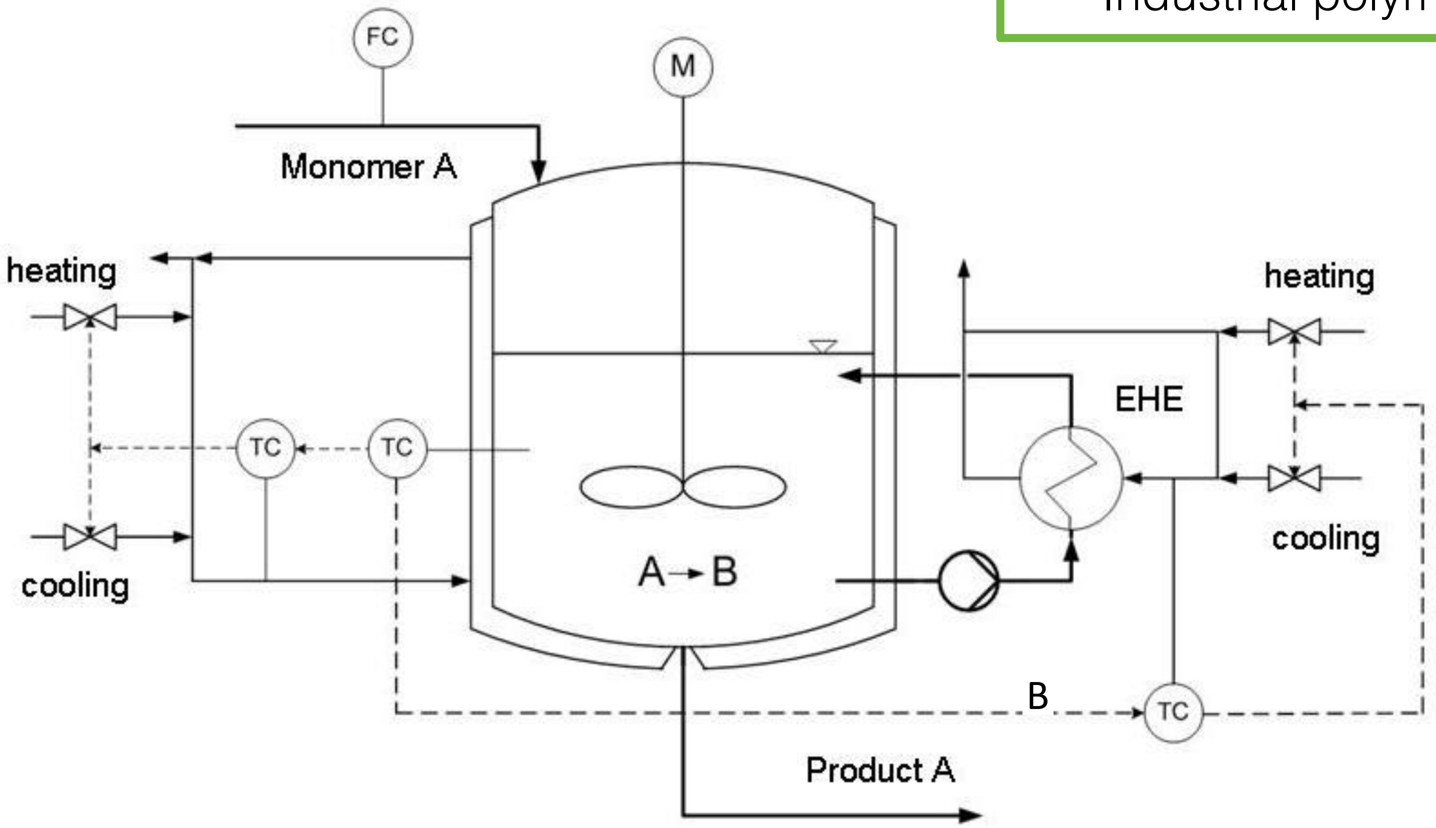
Avoid the exponential growth by branching up to the **robust horizon**

- Afterwards, assume that uncertainty remains constant
- This is a known heuristic in stochastic programming



An example of model predictive control

Industrial polymerization reactor



Standard NMPC: No uncertainties

Control objective:

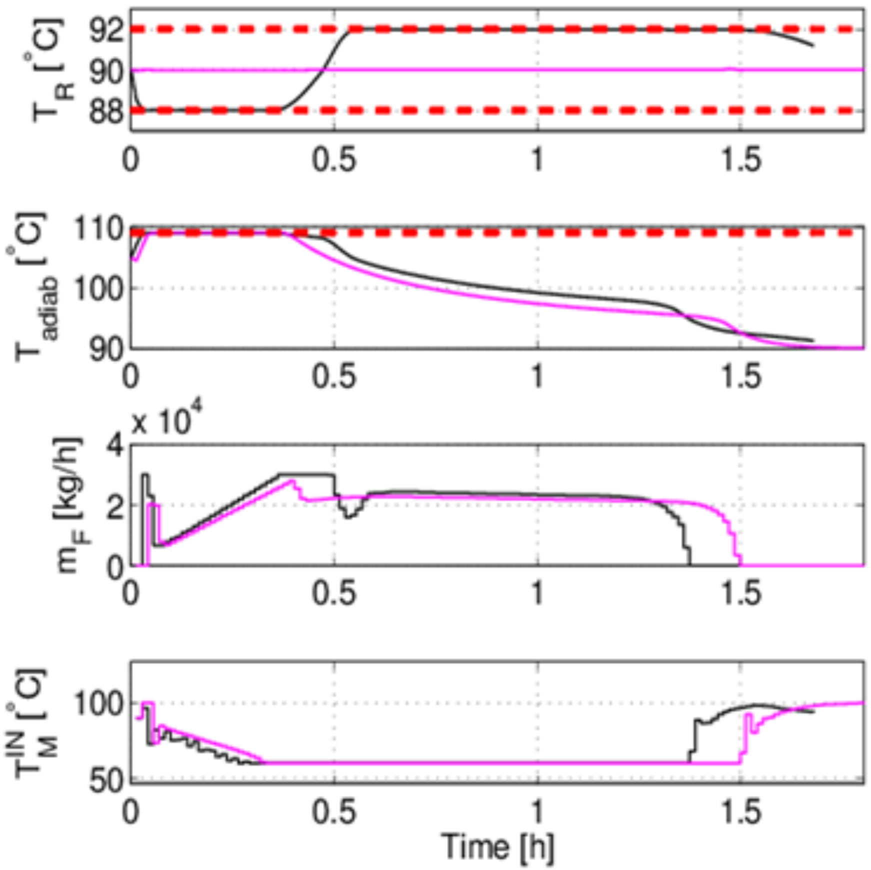
- Minimize batch time:
  - Satisfy quality and safety constraints
  - For all values of the uncertainty (+-30%)

Controller design:

- Tracking or economic cost function
- Prediction horizon of 20 steps

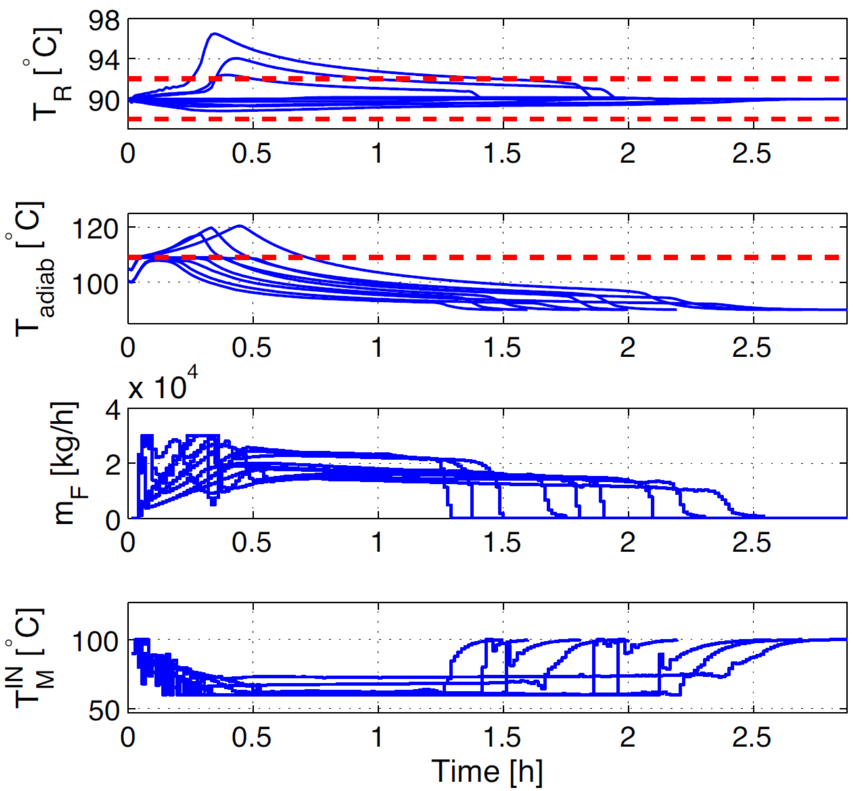
With a **perfect model:**

- Constraints are always satisfied
- Batch time reduced for economic cost



Simulation results for different scenarios

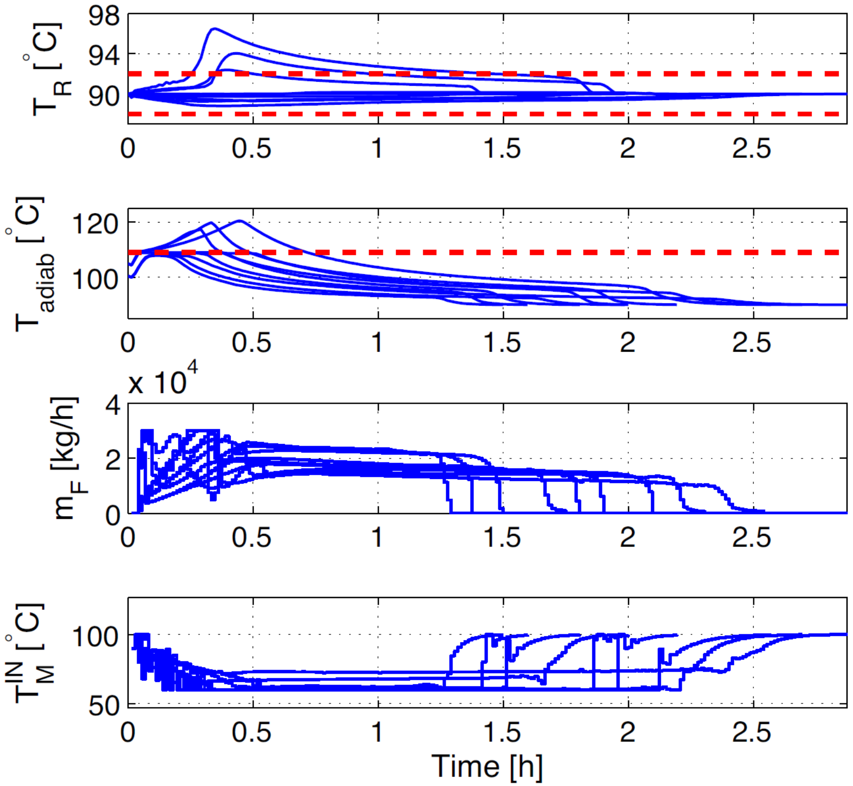
Standard NMPC (tracking)



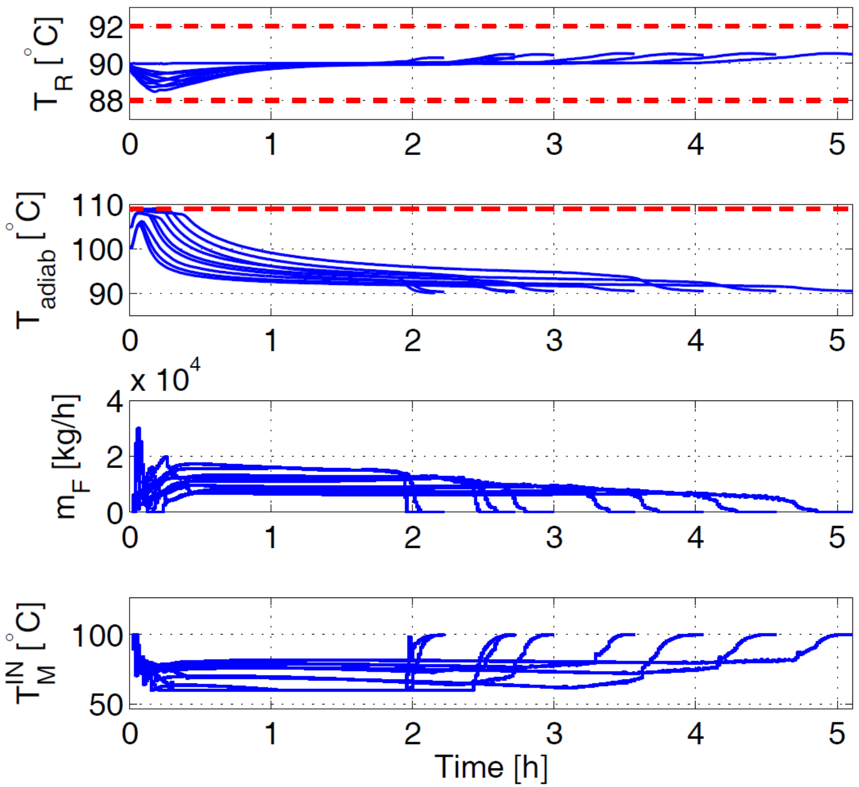
Simulations for different values of k and ΔH ($\pm$30%)

Simulation results for different scenarios

Standard NMPC (tracking)



Standard NMPC with conservative choice of parameters

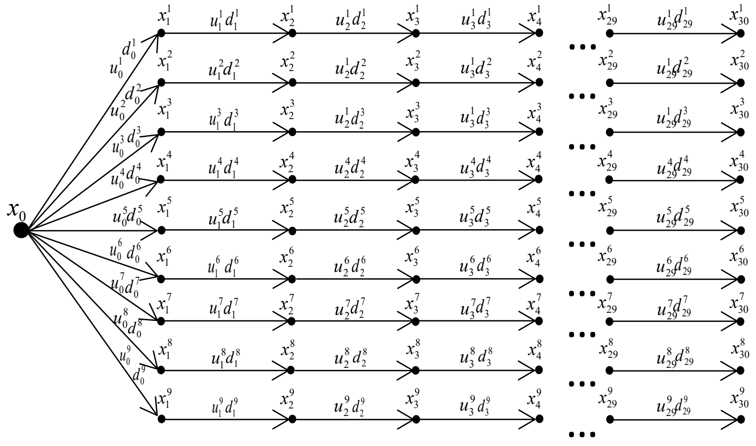


Simulations for different values of k and ΔH ($\pm 30\%$)

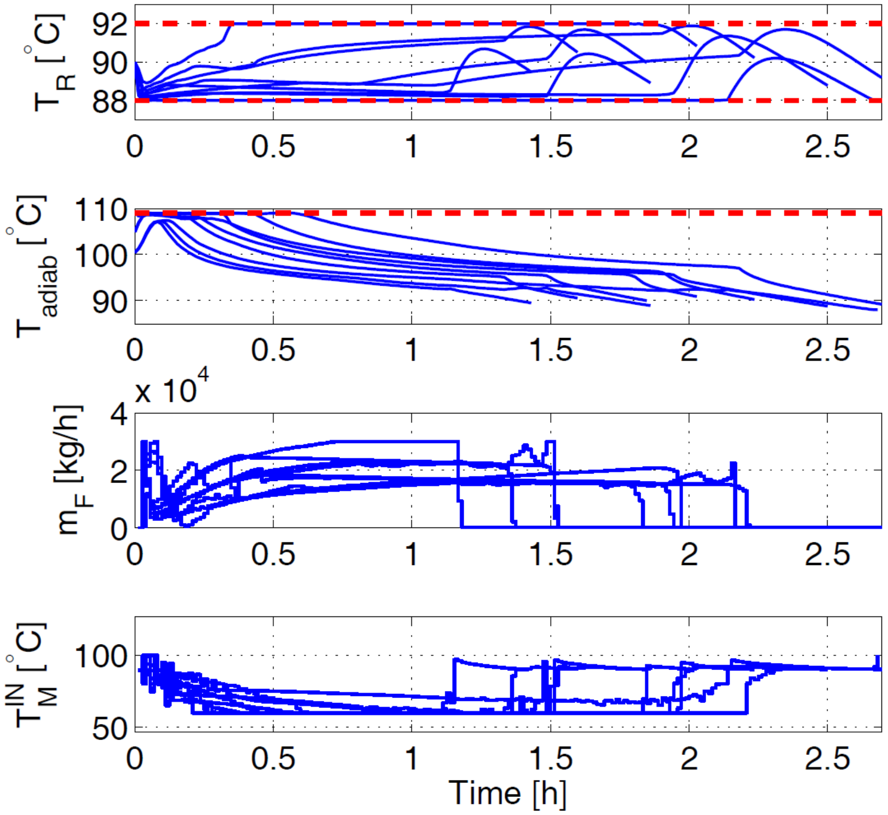
Simulation results for multi-stage NMPC

Simple scenario tree

- Extreme values of the uncertainty
- Branch the tree only one stage
- Economic cost function



Multi-stage NMPC



Simulations for different values of k and ΔH ($\pm$30%)

Comparison with standard NMPC

[Lucia, Andersson, Brand, Diehl and Engell, Journal of Process Control, 2014]

Performance

Scenario		Batch time in hours		
ΔH_R	k_0	Standard NMPC	Standard (c.c.)	Multi-stage
+30%	+30%	infeasible	2.15	2.03
+30%	0%	infeasible	2.72	2.24
+30%	-30%	infeasible	4.05	2.69
0%	+30%	1.60	2.22	1.60
0%	0%	1.81	3.00	1.84
0%	-30%	2.69	4.57	2.50
-30%	+30%	1.50	2.72	1.43
-30%	0%	1.99	3.57	1.86
-30%	-30%	2.88	5.11	2.68
Av. batch time [h]		infeasible	3.35	2.10

Batch time reduction of 60% w.r.t. standard (c.c.) NMPC

Comparison with standard NMPC

[Lucia, Andersson, Brand, Diehl and Engell, Journal of Process Control, 2014]

Performance

Scenario		Batch time in hours		
ΔH_R	k_0	Standard NMPC	Standard (c.c.)	Multi-stage
+30%	+30%	infeasible	2.15	2.03
+30%	0%	infeasible	2.72	2.24
+30%	-30%	infeasible	4.05	2.69
0%	+30%	1.60	2.22	1.60
0%	0%	1.81	3.00	1.84
0%	-30%	2.69	4.57	2.50
-30%	+30%	1.50	2.72	1.43
-30%	0%	1.99	3.57	1.86
-30%	-30%	2.88	5.11	2.68
Av. batch time [h]		infeasible	3.35	2.10

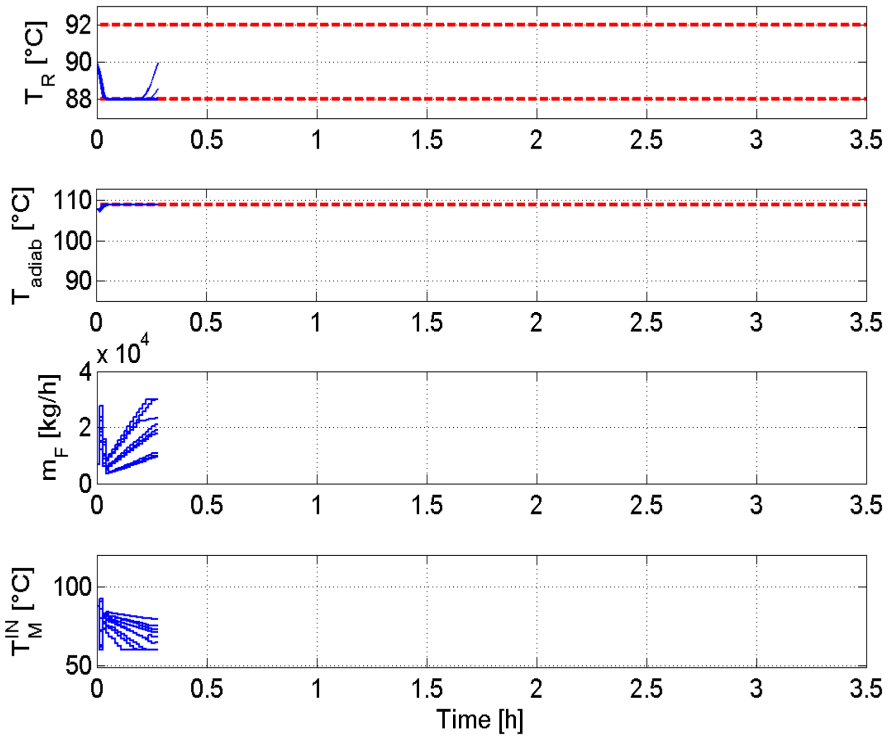
Computation times

Comp time [s]	Standard NMPC	Standard (c.c.)	Multi-stage
Average	0.072	0.059	1.134
Maximum	0.230	0.179	1.550

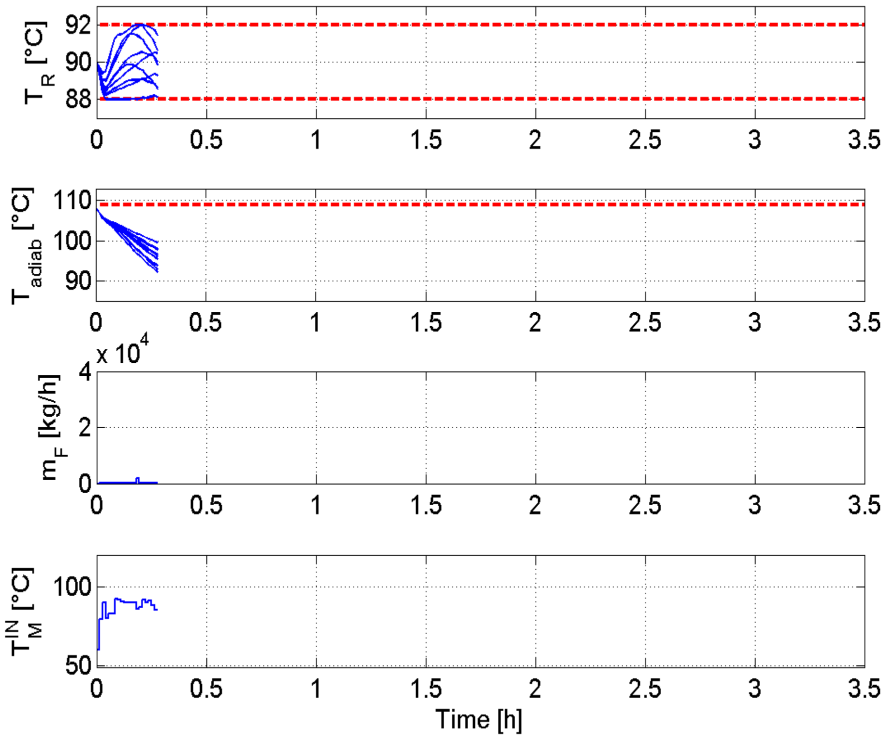
Batch time reduction of 60% w.r.t. standard (c.c.) NMPC

Comparison with open-loop robust NMPC

Multi-stage NMPC



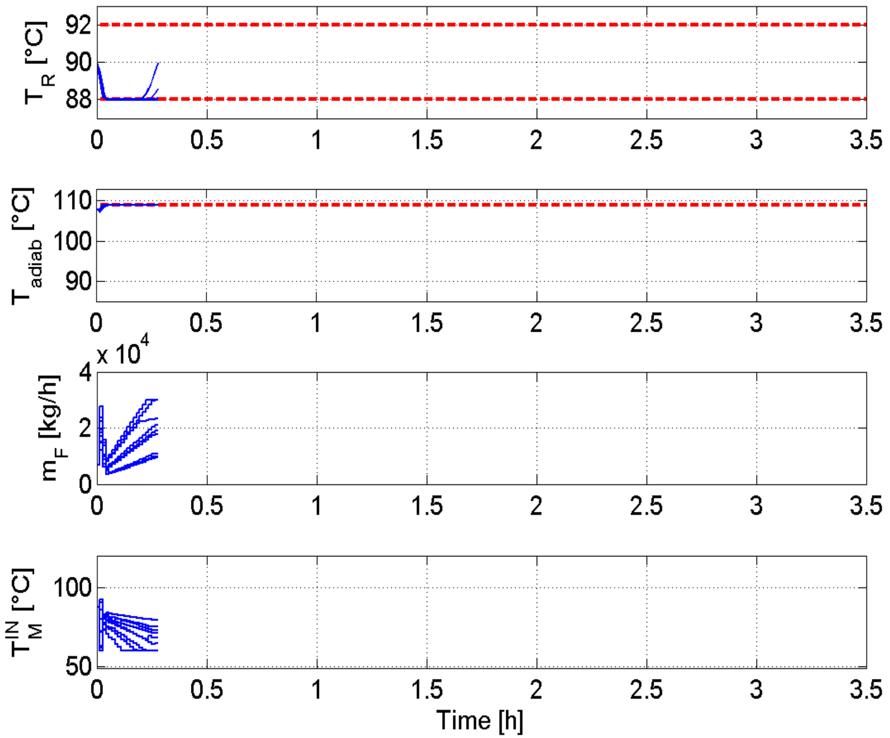
Open-loop robust NMPC



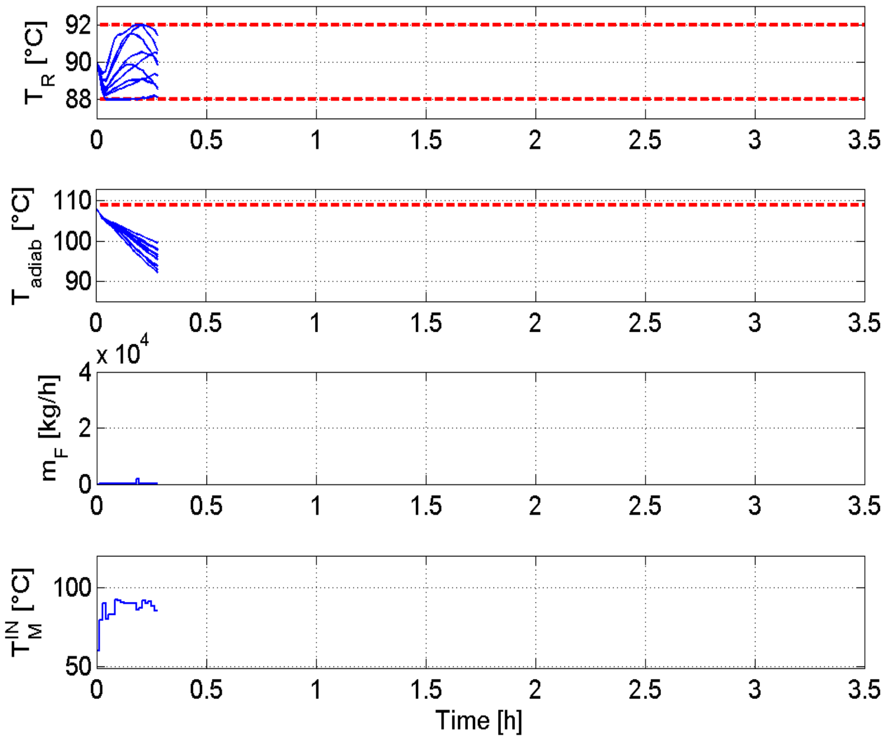
Comparison with open-loop robust NMPC

25% longer batch!

Multi-stage NMPC



Open-loop robust NMPC



Scenarios: 9
Variables: 7,700
Average CPU time: 1.10 s

Scenarios: 81
Variables: 67,000
Average CPU time: 14.1 s

Scenarios: 729
Variables: 600,000
Average CPU time: 240 s

A widely used approach for robust NMPC

- Many applications reported in literature
  - Bioethanol production [Skupin et al, Computers and Chemical Engineering, 2022]
  - Pumping treatment in hydraulic fracturing [Lin, Eason and Biegler, AIChE Journal 2022]
  - Artificial pancreas system [Colmegna et al., CDC 2022]
  - Gas Network economic MPC [Naik, Ghilardi, Parker and Biegler, Chemical Engineering Science, 2025]
- Many possible extensions:
  - Guarantees for nonlinear systems
    [Karg, Alamo and Lucia, Int. Journal of Robust and Nonlinear Control, 2021] [Lucia, Paulen, Engell, CDC 2014]
  - Simplifications of the scenario tree using monotonicity of systems
    [Heinlein, Subramanian, Lucia, TAC 2025]
  - Sensitivity-based scenario selection, decomposition algorithms
    [Thombre, Yu, Jäschke, Biegler, CACE, 2021] [D. Krishnamoorthy, B. Foss, S. Skogestad, JPC, 2019]
  - Generation of scenarios via bayesian neural networks
    [Bao, Chan, Mesbah and Velni, International Journal of Robust and Nonlinear Control, 2022]

Tube-based MPC

A simple robust MPC scheme (especially for linear systems)

Tube-based model predictive control

Consider the linear system $x^+ = Ax + Bu + w$ with $w \in \mathbb{W}$

The **nominal system** is: $z^+ = Az + Bu$

The deviation of the real system from the nom. system is: $e = x - z$

The deviation satisfies the dynamics: $e^+ = Ae + w$

And the error at step i can be computed as:

$$e(i) = A^i e(0) + \sum_{j=0}^{i-1} A^j w(j)$$

[Mayne et al., 2005]
[Rawlings, Mayne, Diehl, 2017]
[Kouvaritakis and Canon, 2016]

Tube-based model predictive control

The uncertainty sets at each step $S(i)$ are defined as:

$$S(i) := \sum_{j=0}^{i-1} A^j \mathbb{W} = \mathbb{W} \oplus A\mathbb{W} \oplus \cdots \oplus A^{i-1}\mathbb{W}$$

Definitions

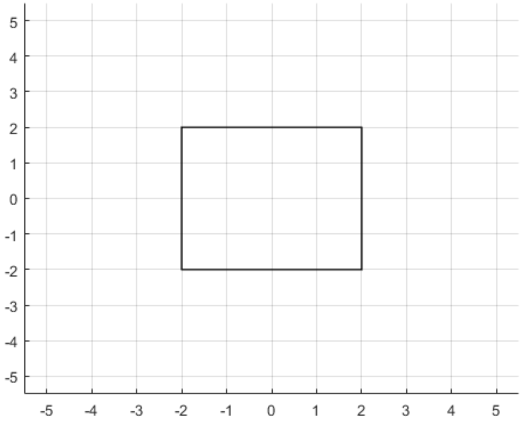
Set addition (Minkowski sum): $\mathbb{A} \oplus \mathbb{B} := \{a + b \,|\, a \in \mathbb{A}, b \in \mathbb{B}\}$

Set substraction (Minkowski difference):

$$\mathbb{A} \ominus \mathbb{B} := \{x \in \mathbb{R}^n \,|\, \{x\} \oplus \mathbb{B} \subseteq \mathbb{A}\}$$

Set operations

$$\mathbb{A} \oplus \mathbb{B} := \{a + b \,|\, a \in \mathbb{A}, b \in \mathbb{B}\}$$



Set $\mathbb{A}$

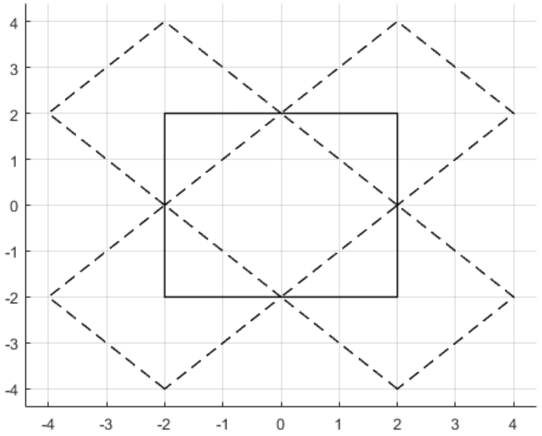
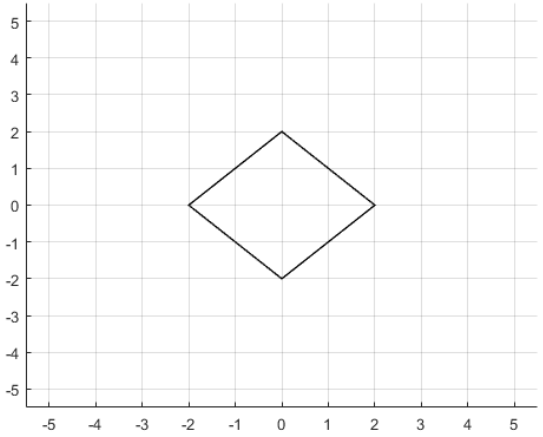
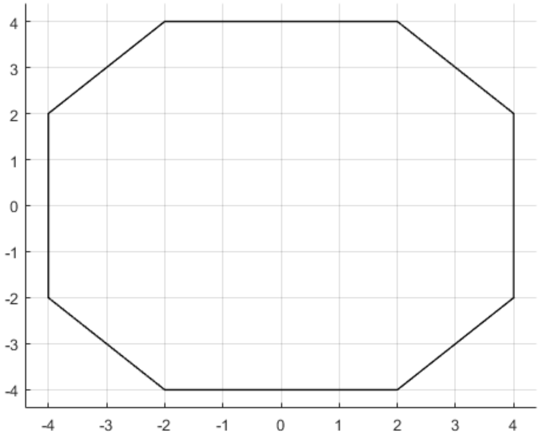


Illustration of Minkowski sum



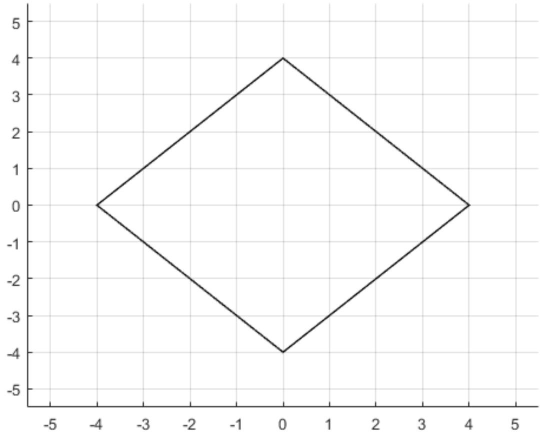
Set $\mathbb{B}$



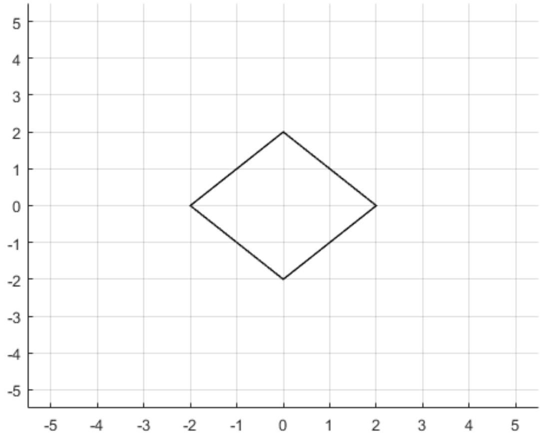
Set $\mathbb{A} \oplus \mathbb{B}$

Set operations

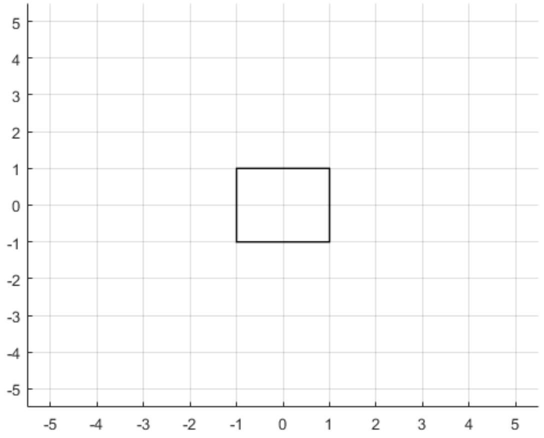
$$\mathbb{A} \ominus \mathbb{B} := \{x \in \mathbb{R}^n \,|\, \{x\} \oplus \mathbb{B} \subseteq \mathbb{A}\}$$



Set $\mathbb{A}$



Set $\mathbb{A} \ominus \mathbb{B}$



Set $\mathbb{B}$

Note: $(\mathbb{A} \oplus \mathbb{B}) \ominus \mathbb{B} \neq \mathbb{A}$

Tube-based model predictive control

Tube generated by the open-loop control sequence

$$X(0; x_0) = \{x_0\} \qquad X(i; x_0, \mathbf{u}) = \{z(i)\} \oplus S(i)$$

Tube-based model predictive control

Tube generated by the open-loop control sequence

$$X(0; x_0) = \{x_0\} \qquad X(i; x_0, \mathbf{u}) = \{z(i)\} \oplus S(i)$$

If A is stable, then $S(\infty) := \sum_{j=0}^{\infty} A^j \mathbb{W}$ exists and is an **outer-bounding tube**

$$\hat{X}(0; x_0) = \{x_0\} \qquad \hat{X}(i; x_0, \mathbf{u}) = \{z(i)\} \oplus S(\infty)$$

Constant cross section

Tube-based model predictive control

How to consider **feedback** in the predictions?

- Use $u = v + K(x - z)$

The real system state satisfies: $x^+ = Ax + Bv + BKe + w$

And the nominal system: $z^+ = Az + Bv$

Tube-based model predictive control

How to consider **feedback** in the predictions?

- Use $u = v + K(x - z)$

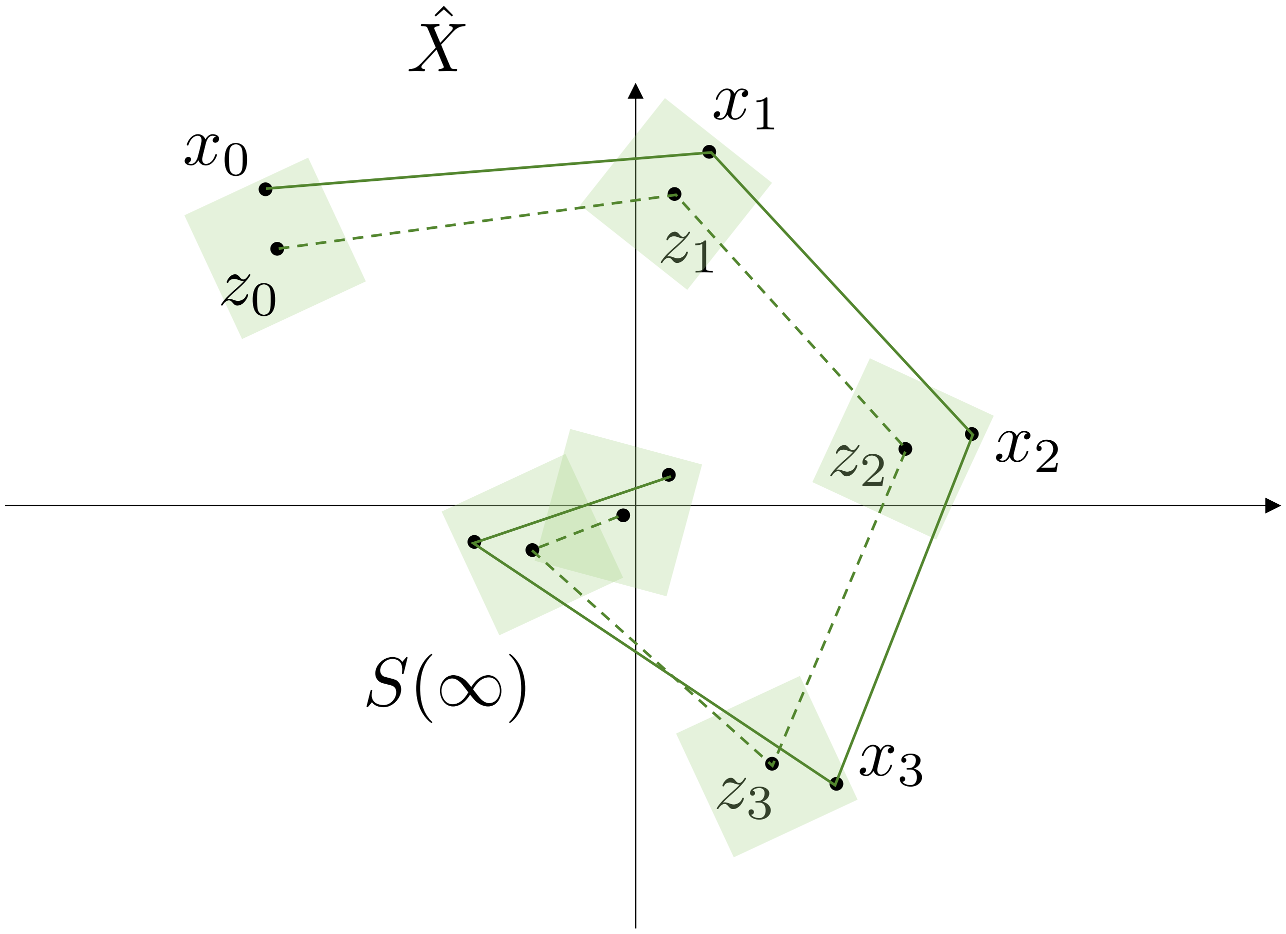
The real system state satisfies: $x^+ = Ax + Bv + BKe + w$

And the nominal system: $z^+ = Az + Bv$

The error dynamics are thus: $e^+ = A_K e + w$

The **uncertainty sets** $S(i)$ **can be made smaller** $S(i) := \sum_{j=0}^{i-1} A_K^j \mathbb{W}$

Tube-based model predictive control



Tube-based model predictive control

How to guarantee constraint satisfaction?

- Impose **tightened constraints for the nominal system**

$$z(i) \in \mathbb{Z} := \mathbb{X} \ominus S(\infty)$$
$$v(i) \in \mathbb{V} := \mathbb{U} \ominus KS(\infty)$$

Stability is guaranteed if the nominal system and A_K are stable

Tube-based MPC: Four main challenges

1. Which objective should be optimized?

   Nominal objective
2. Constraints should be satisfied for all possible outcomes

   Tighten constraints based on invariant sets
3. Performance should not be overly conservative

   Ancillary feedback controller: stay close to nominal traj.
4. The closed-loop should be stable for all possible outcomes

   Stable nominal and error closed-loop systems

A simple example

Robust constraint satisfaction with the complexity of standard MPC

Simple example

Linear system $x(k+1) = Ax(k) + Bu(k) + w(k)$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix} \leq w \leq \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

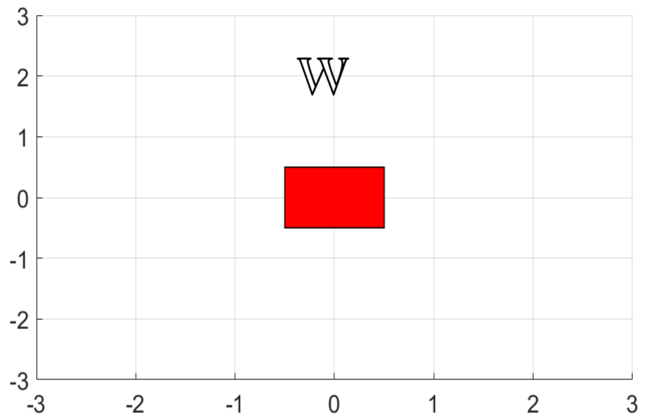
Constraints

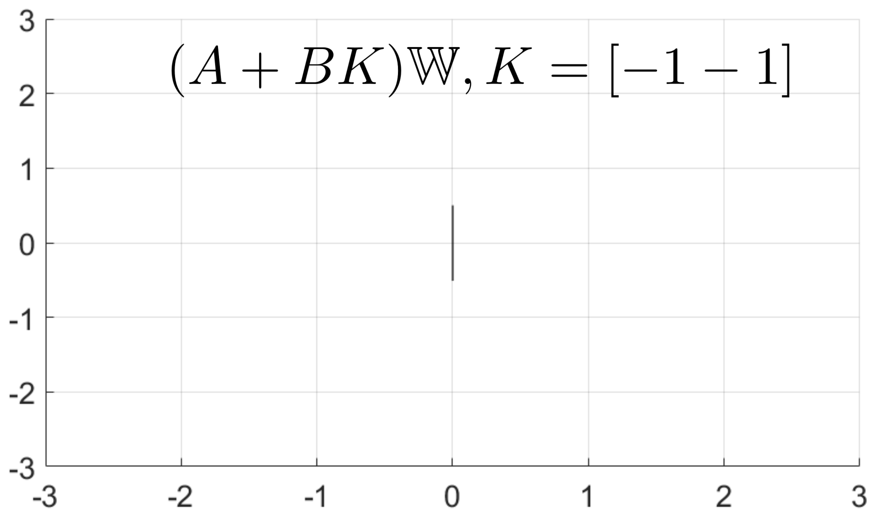
$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 3 \\ 3 \end{bmatrix}, -1 \leq u(k) \leq 1, \forall k \geq 0$$

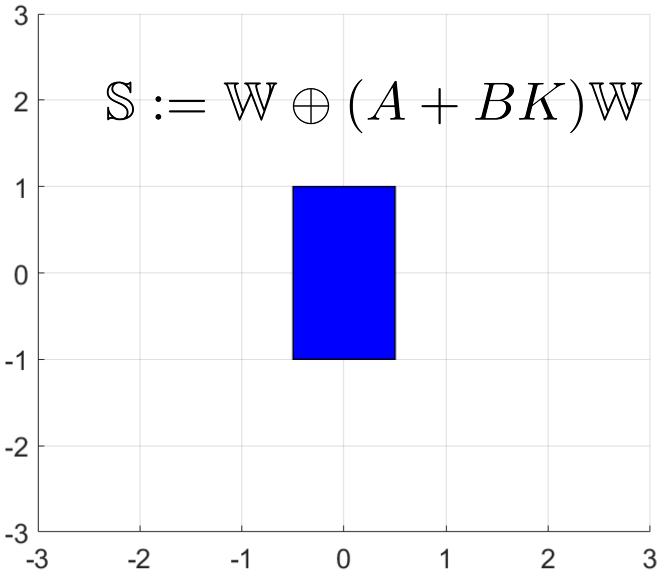
Stage cost

$$x(k)^T Q x(k) + u(k)^T R u(k), Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 0.01$$

Minimal RPI Set

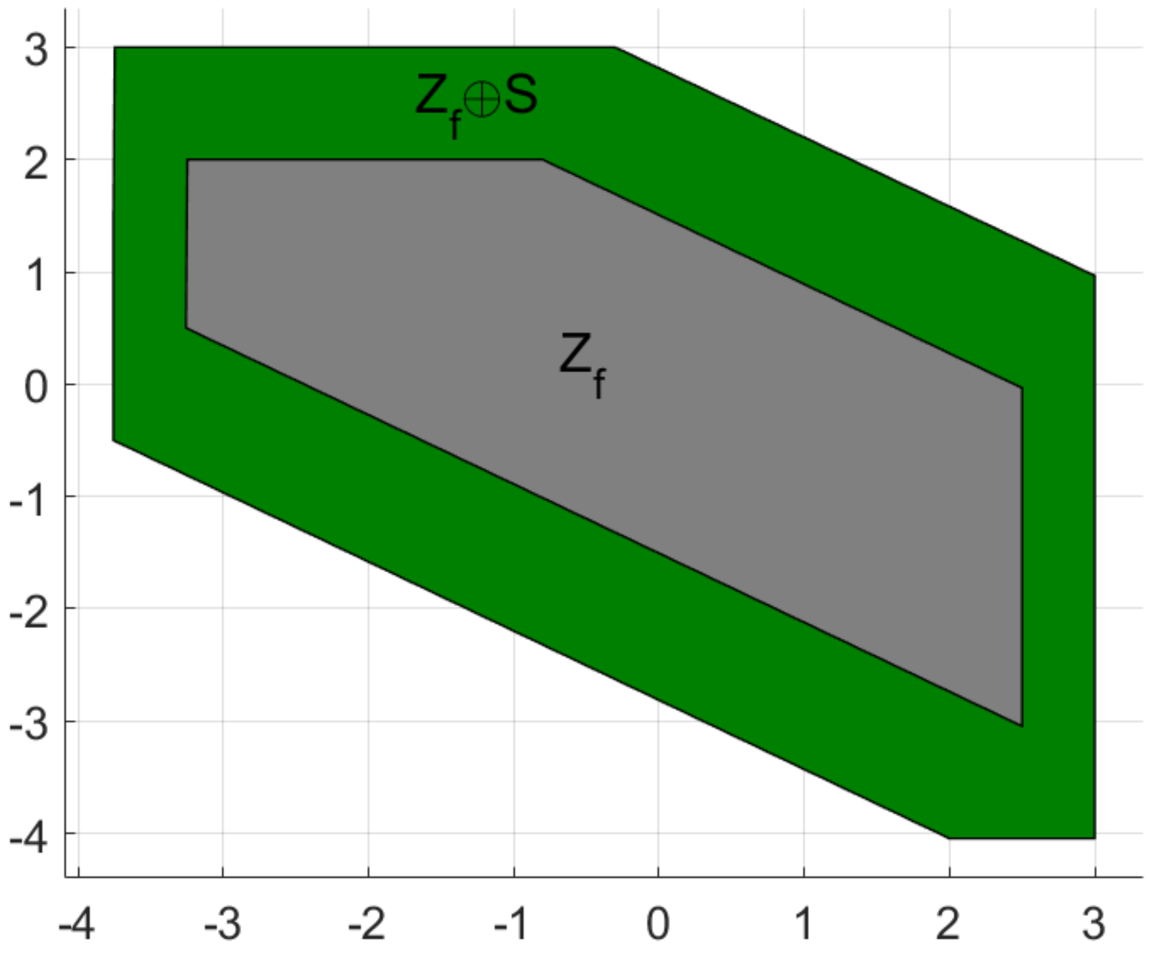




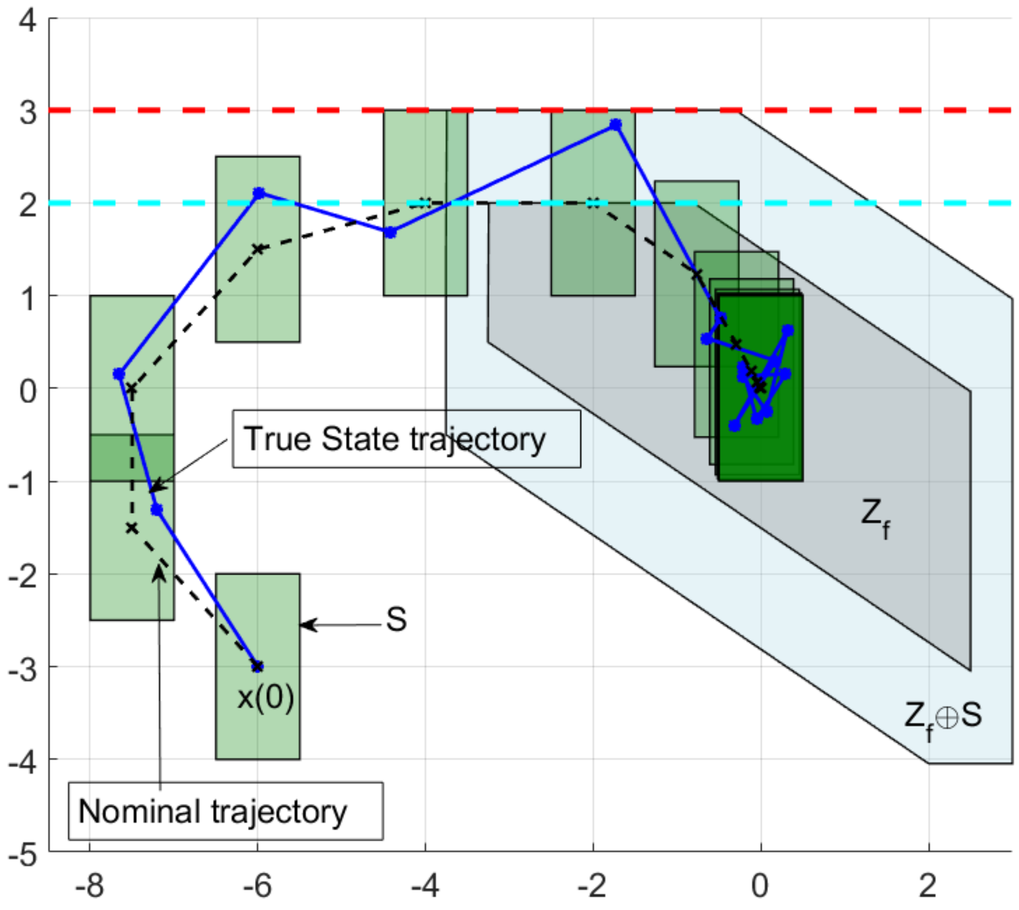


$(A+BK)^2$ has zero in all elements

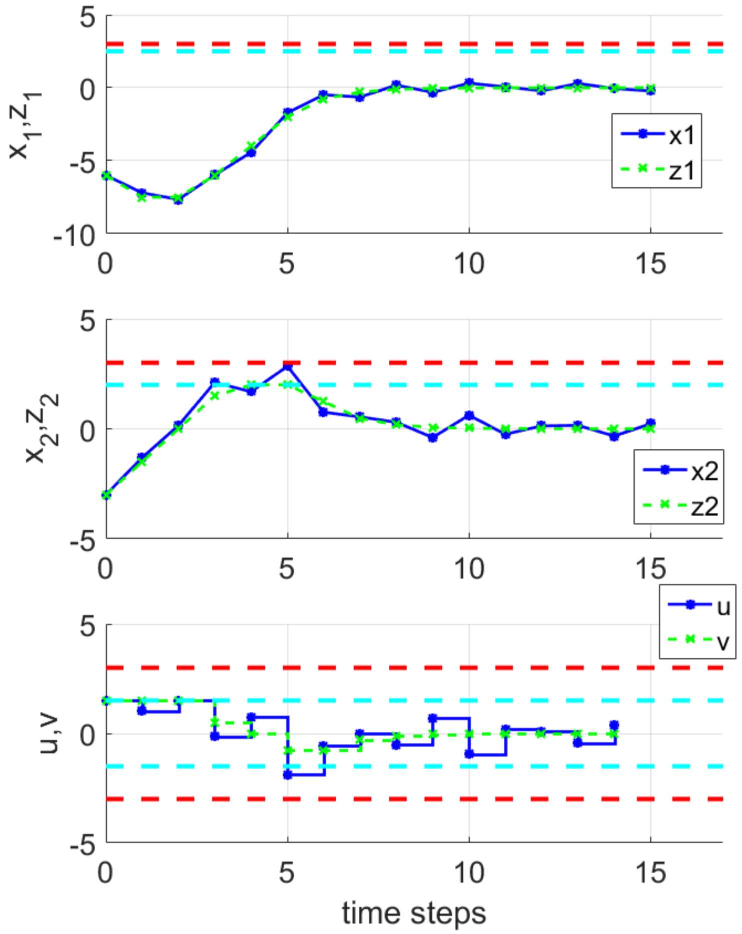
Terminal set obtained using inf. horizon LQR gain



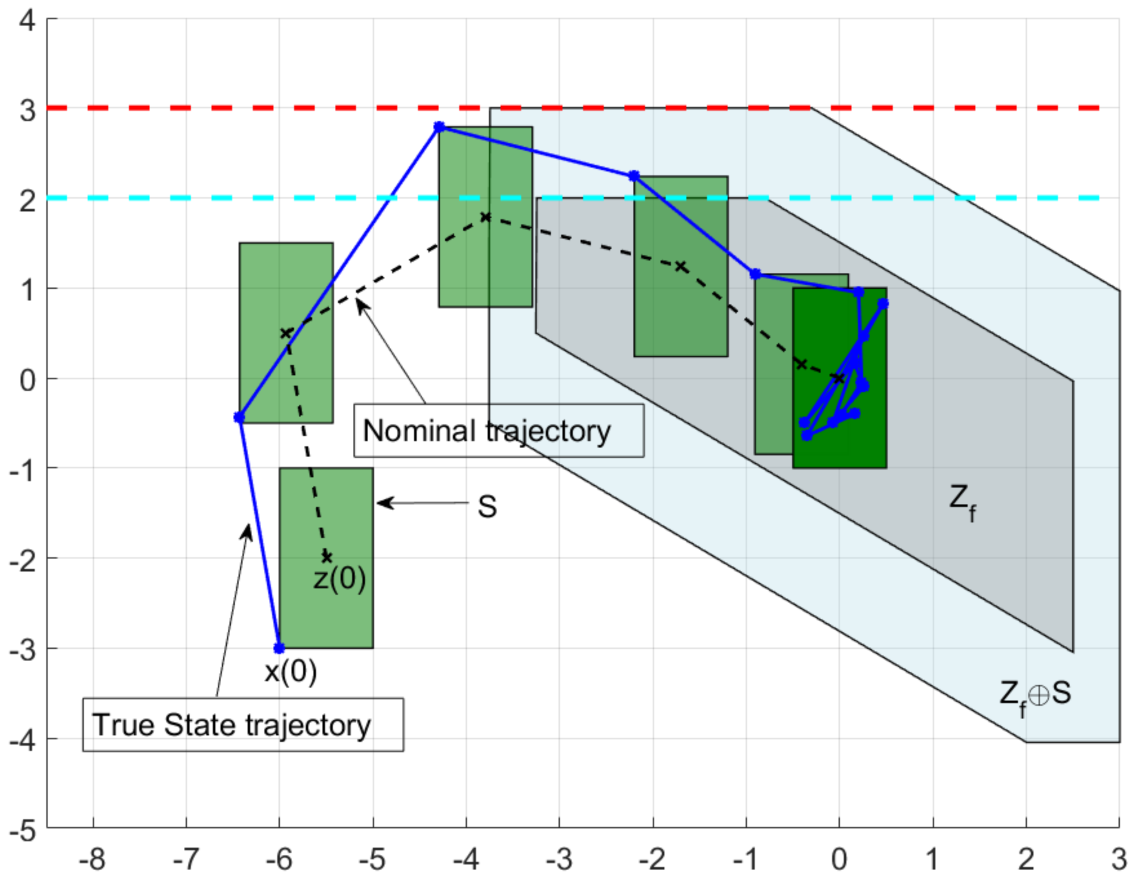
Tube with initial condition fixed



State and control trajectories

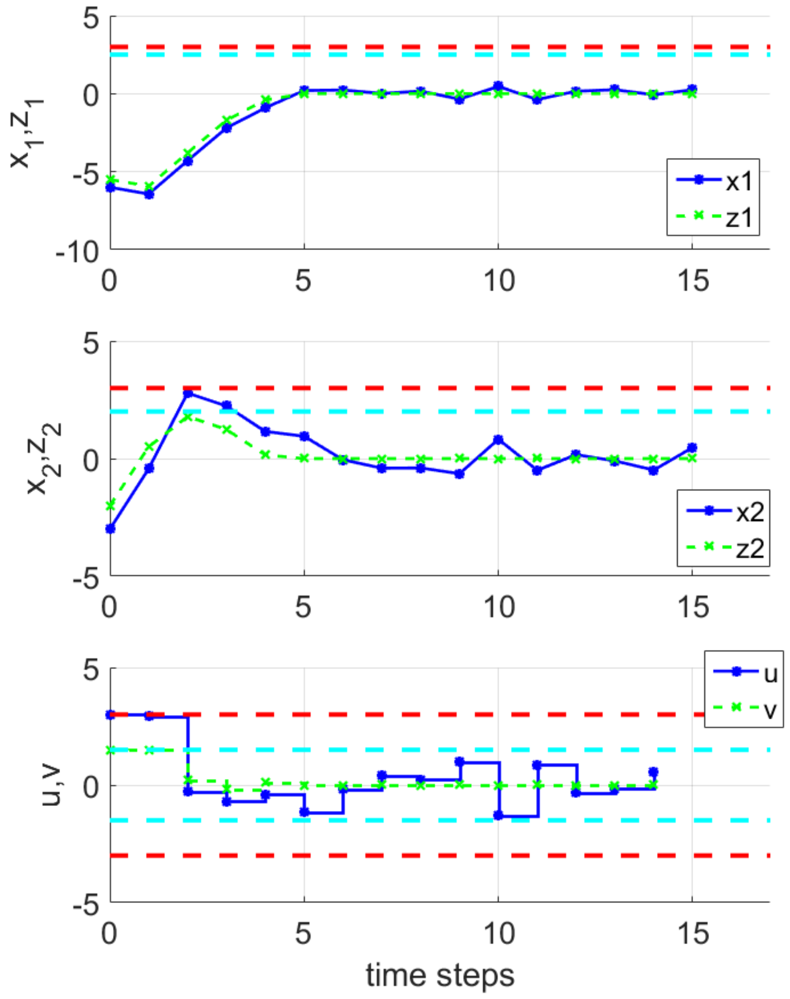


Tube with relaxed initial condition



Faster convergence can be achieved!

State and control trajectories



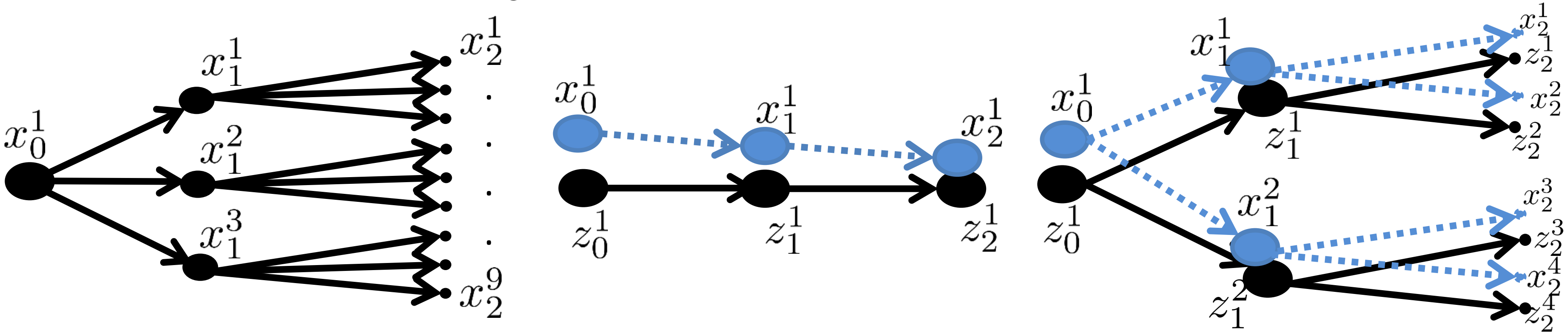
Other approaches

- General complexity and homothetic tube-based MPC
  - Use a more complex description of the uncertainty sets to improve performance

    [Kouvaritakis and Canon, 2016]

- Tube-enhanced multi-stage MPC
  - Use multi-stage MPC for *significant* uncertainties and use tube-based MPC for *less important* ones to avoid too large scenario trees

    [Subramanian, Lucia, Paulen and Engell. International Journal of Robust and Nonlinear Control, 2021]



Multi-stage NMPC Tube-based NMPC [Mayne et al., 2011] TEMS NMPC

What can a scenario tree give you?

- Linear feedback policies are tractable but suboptimal even in the constrained linear MPC case
- Multi-stage MPC can lead to larger domain of attractions
- Example for a linearized CSTR [Subramanian, Lucia, Paulen and Engell. Int. Journal of RNC, 2021]