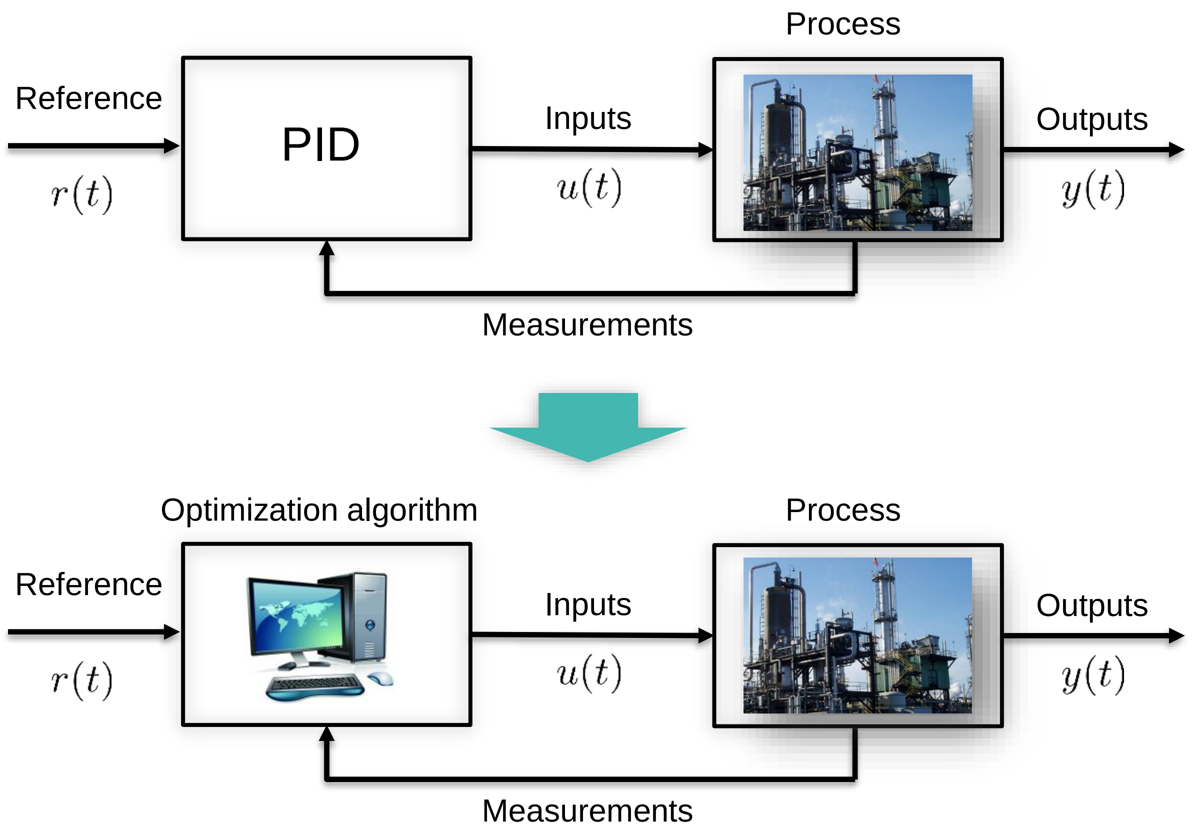
NTNU | Norwegian University of Science and Technology

TTK4135 – Lecture 10
Model Predictive Control

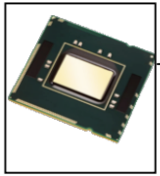
Lecturer: Lars Imsland

Model Predictive Control – control based on optimization



A model of the process is used to compute the control signals (inputs) that optimize predicted future process behavior

MPC: Applications



Computer control — ns

μs — Power systems

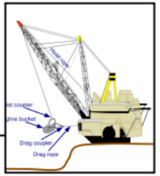
Traction control — ms

Seconds — Buildings

Refineries — Minutes

Hours — Nurse rostering

Train scheduling — Days

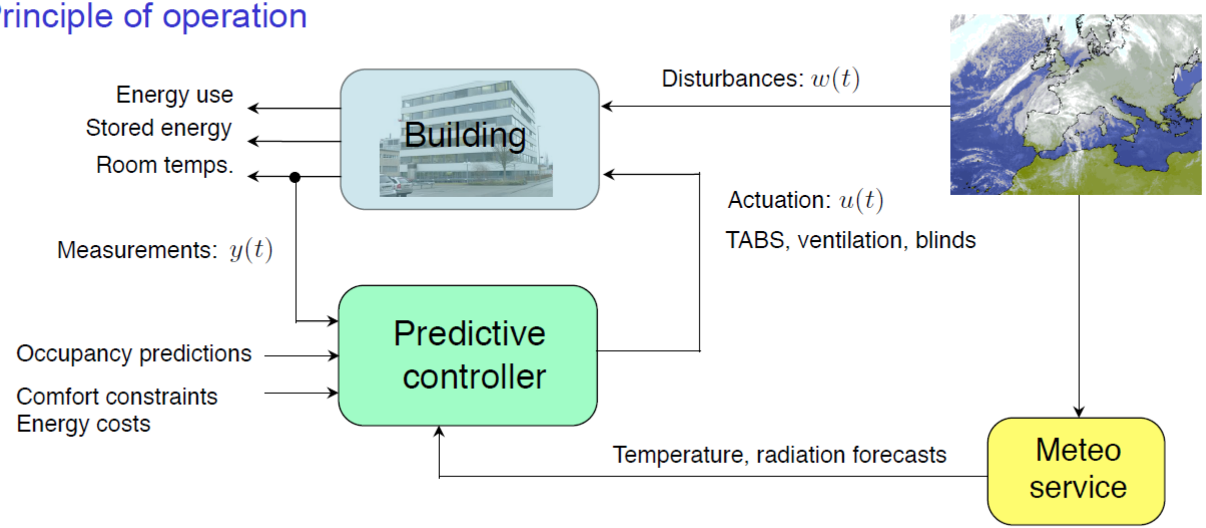


Weeks — Production planning

Zeilinger, Jones, Borrelli, Morari (ETH)

Model predictive control (MPC)

Principle of operation



$$\text{Predicted Cost} \quad = \quad \underset{u(t)}{\text{minimize}} \quad \text{Expected} \left(\sum_{t}^{t+N} \text{energy cost(t)} \right)$$

Minimize the predicted energy cost

$$\begin{aligned} \text{subject to} \quad & u(t) \in \mathcal{U} \\ & x(t) \in \mathcal{X} \\ & x(t+1) = f(x(t), u(t), w(t)) \end{aligned}$$

Actuation within limits

Predicted temperatures within limits

Predicted dynamics of the building

From ETH

Open-loop optimization with linear state-space model

QP

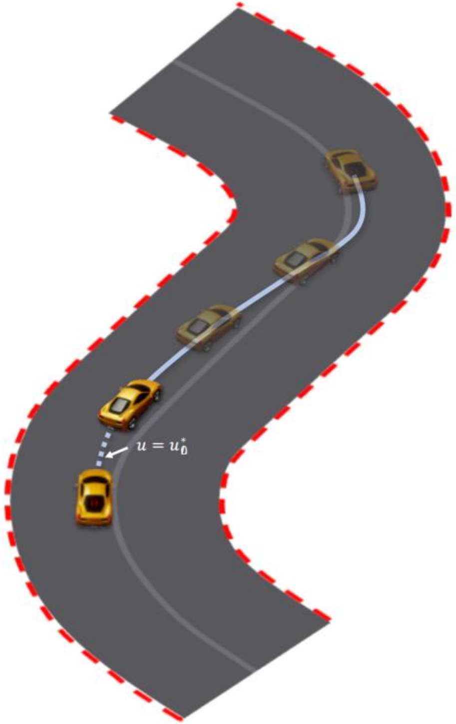
$$\min_{z\in\mathbb{R}^n} f(z) = \sum_{t=0}^{N-1} \frac{1}{2}x_{t+1}^\top Q_{t+1}x_{t+1} + d_{x,t+1}x_{t+1} + \frac{1}{2}u_t^\top R_t u_t + d_{u,t}u_t + \frac{1}{2}\Delta u_t^\top S \Delta u_t$$

subject to

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t, \quad t = \{0,\dots,N-1\} \\ x^{\text{low}} &\le x_t \le x^{\text{high}}, \quad t = \{1,\dots,N\} \\ u^{\text{low}} &\le u_t \le u^{\text{high}}, \quad t = \{0,\dots,N-1\} \\ -\Delta u^{\text{high}} &\le \Delta u_t \le \Delta u^{\text{high}}, \quad t = \{0,\dots,N-1\} \end{aligned}$$

where

$$\begin{aligned} &x_0 \text{ and } u_{-1} \text{ is given} \\ \Delta u_t &:= u_t - u_{t-1} \\ z^\top &:= (u_0^\top, x_1^\top, \dots, u_{N-1}^\top, x_N^\top) \\ n &= N \cdot (n_x + n_u) \\ Q_t &\succeq 0 \quad t = \{1,\dots,N\} \\ R_t &\succeq 0 \quad t = \{0,\dots,N-1\} \end{aligned}$$



Open-loop dynamic optimization problem as QP

$$\min_{z \in \mathbb{R}^n} f(z) = \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t$$

subject to

$$\begin{aligned} x_{t+1} &= A x_t + B u_t, \quad t = \{0, \dots, N-1\} \\ x^{\text{low}} &\leq x_t \leq x^{\text{high}}, \quad t = \{1, \dots, N\} \\ u^{\text{low}} &\leq u_t \leq u^{\text{high}}, \quad t = \{0, \dots, N-1\} \end{aligned}$$

where

$$\begin{aligned} &x_0 \text{ is given} \\ &z^\top := (u_0^\top, x_1^\top, \dots, u_{N-1}^\top, x_N^\top) \\ &Q \succeq 0, \quad R \succ 0 \end{aligned}$$

Model predictive control principle

Optimization algorithm

Reference

$$
\begin{aligned}
\min_{z} \quad & \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \ldots, N-1 \\
& x^{\text{low}} \leq x_t \leq x^{\text{high}}, \quad t = 1, \ldots, N \\
& u^{\text{low}} \leq u_t \leq u^{\text{high}}, \quad t = 0, \ldots, N-1 \\
& x_0 = x(t)
\end{aligned}
$$

Inputs

$u(t) = u_0^*$

Process

Outputs

$y(t)$

Plant state $x(t)$

At each sample time:

- Measure or estimate current state $x(t)$
- Find optimal open-loop input sequence $U^* = \left(u_0^*, u_1^*, \ldots, u_{N-1}^*\right)$
- Implement only the first element of sequence: $u(t) = u_0^*$

Model predictive control principle

Optimization algorithm

$$\begin{aligned}\min_{z} \ & \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^{\top} Q x_{t+1} + \frac{1}{2} u_t^{\top} R u_t \\ \text{s.t.} \ & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \ldots, N-1 \\ & x^{\text{low}} \le x_t \le x^{\text{high}}, \quad t = 1, \ldots, N \\ & u^{\text{low}} \le u_t \le u^{\text{high}}, \quad t = 0, \ldots, N-1 \\ & x_0 = x(t)\end{aligned}$$

Reference

Inputs

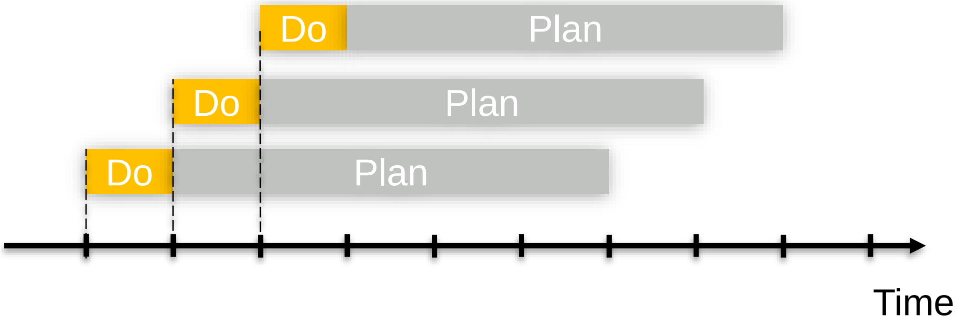
$u(t) = u_0^*$

Process

Outputs

$y(t)$

Plant state $x(t)$



Why? This introduces feedback!

Chess analogy

(The process/disturbance)

The opponent, choosing a counter move

Carlsen's move

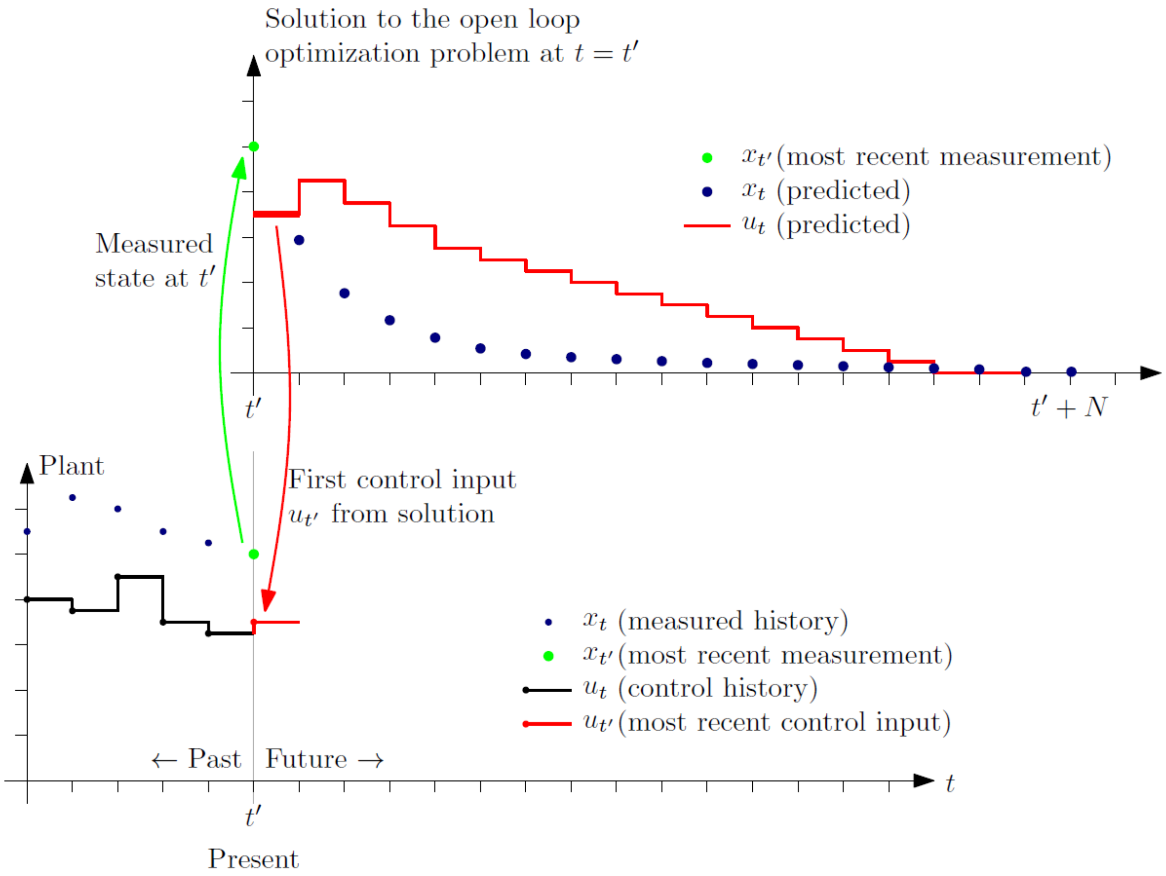
Opponent's move

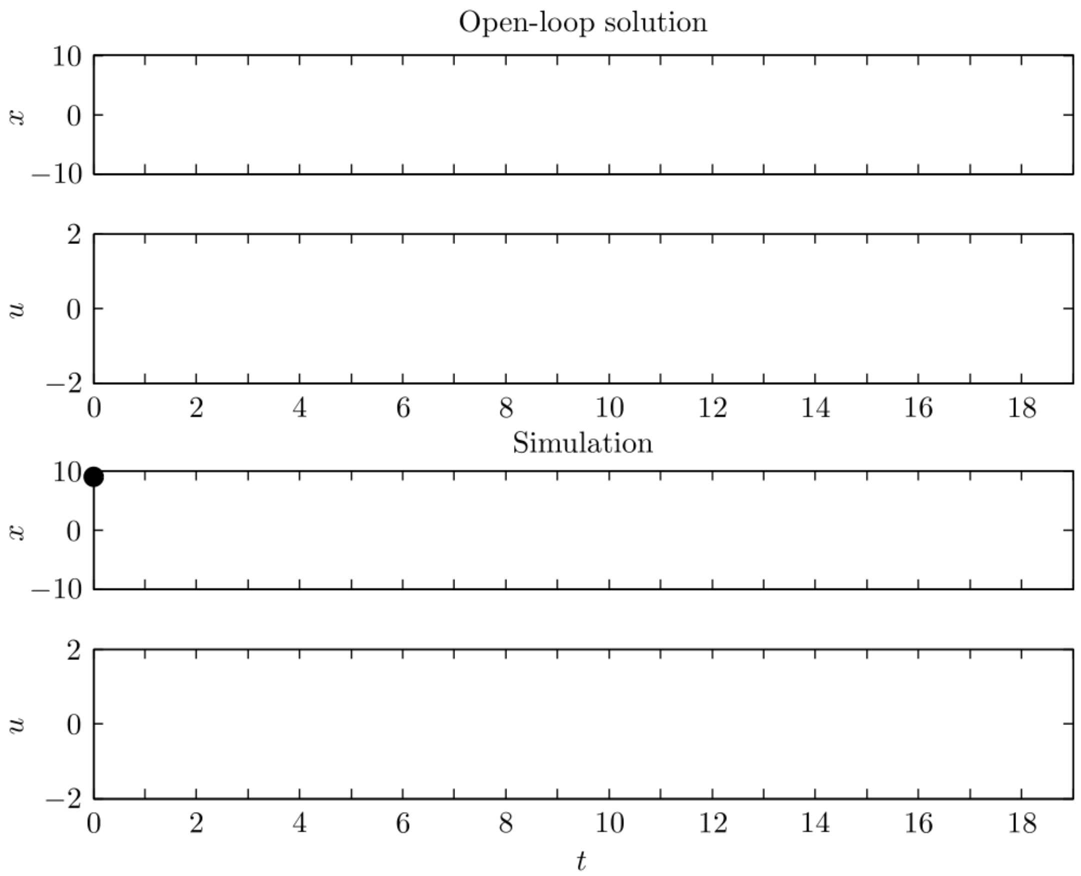
Carlsen, planning many moves into the future

(The MPC controller)

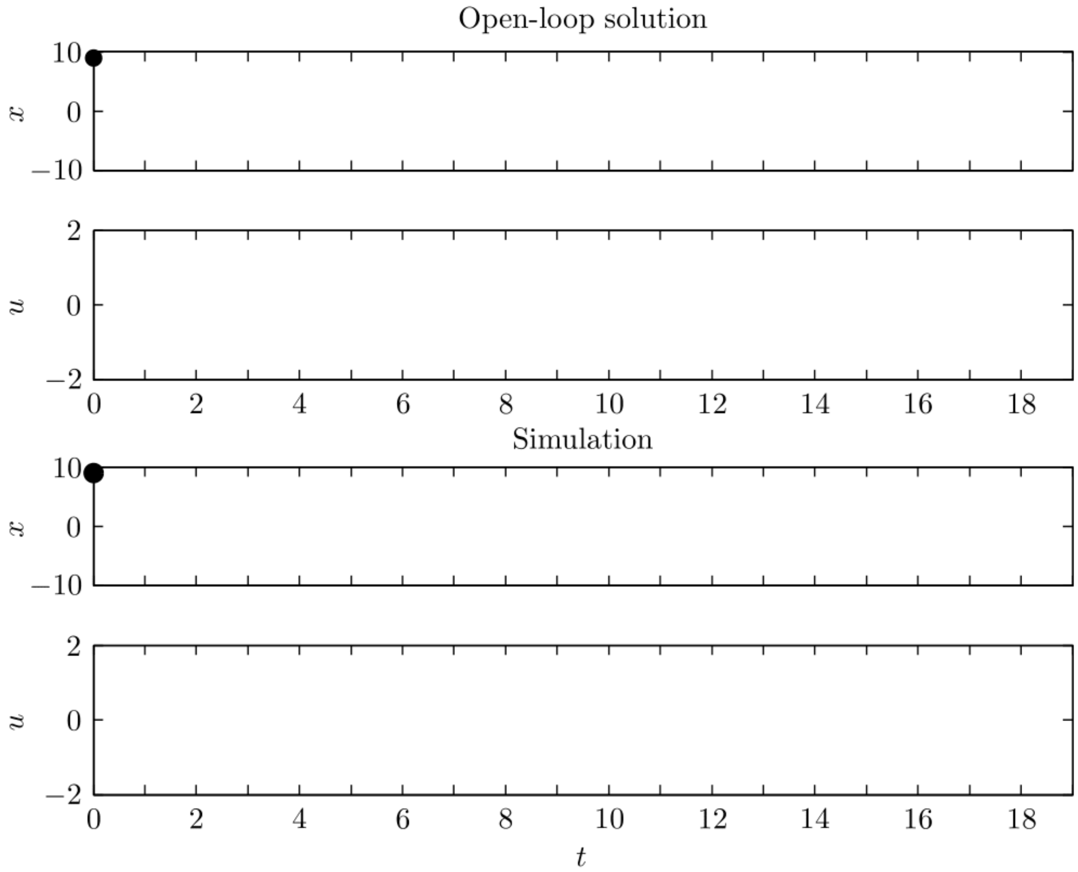
Carlsen vs Karjakin, 2016. Carlsen's (white) move, forced mate in two.

Model predictive control principle

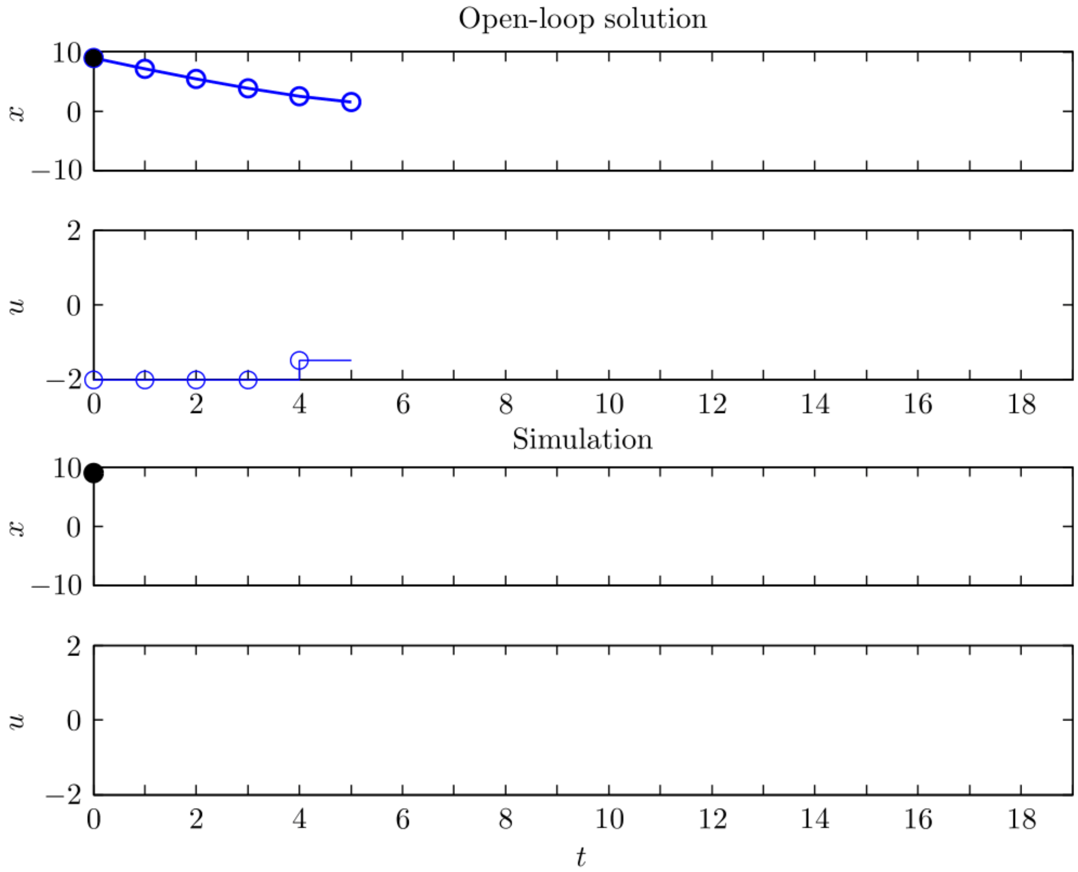




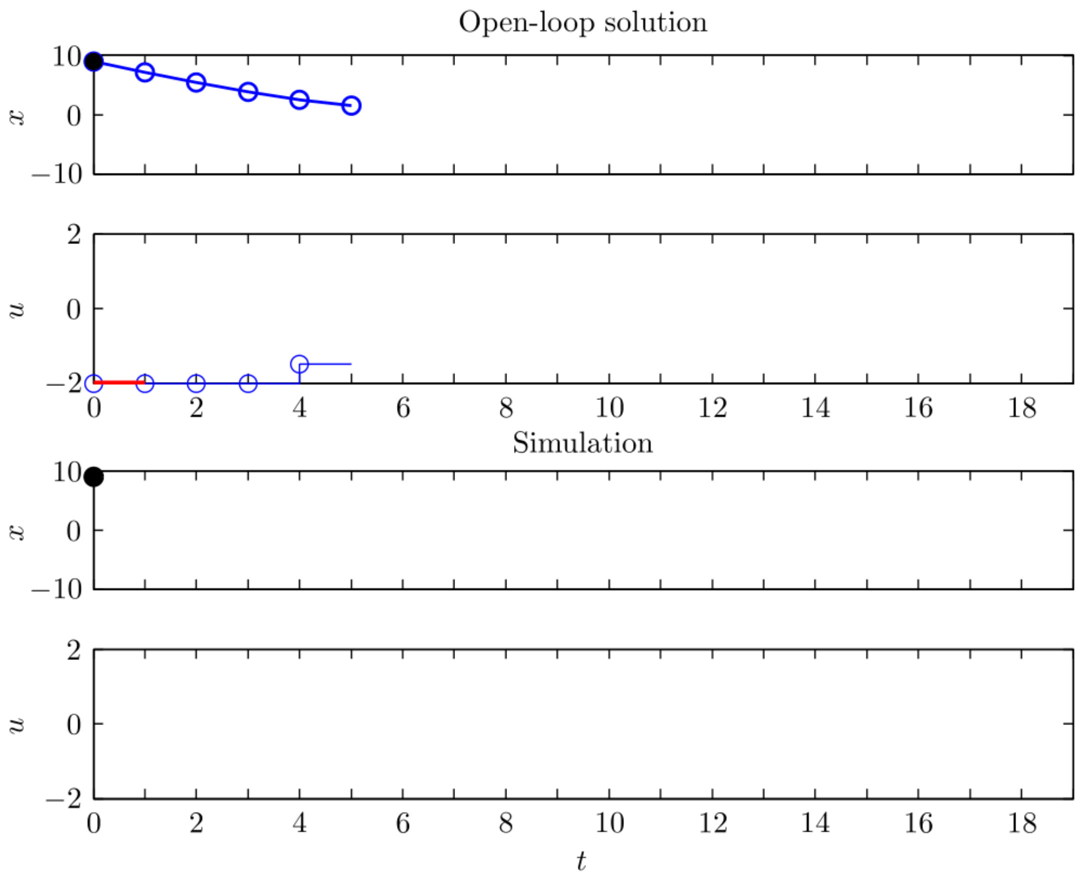
Open-loop solution
x
u
Simulation
x
u
t



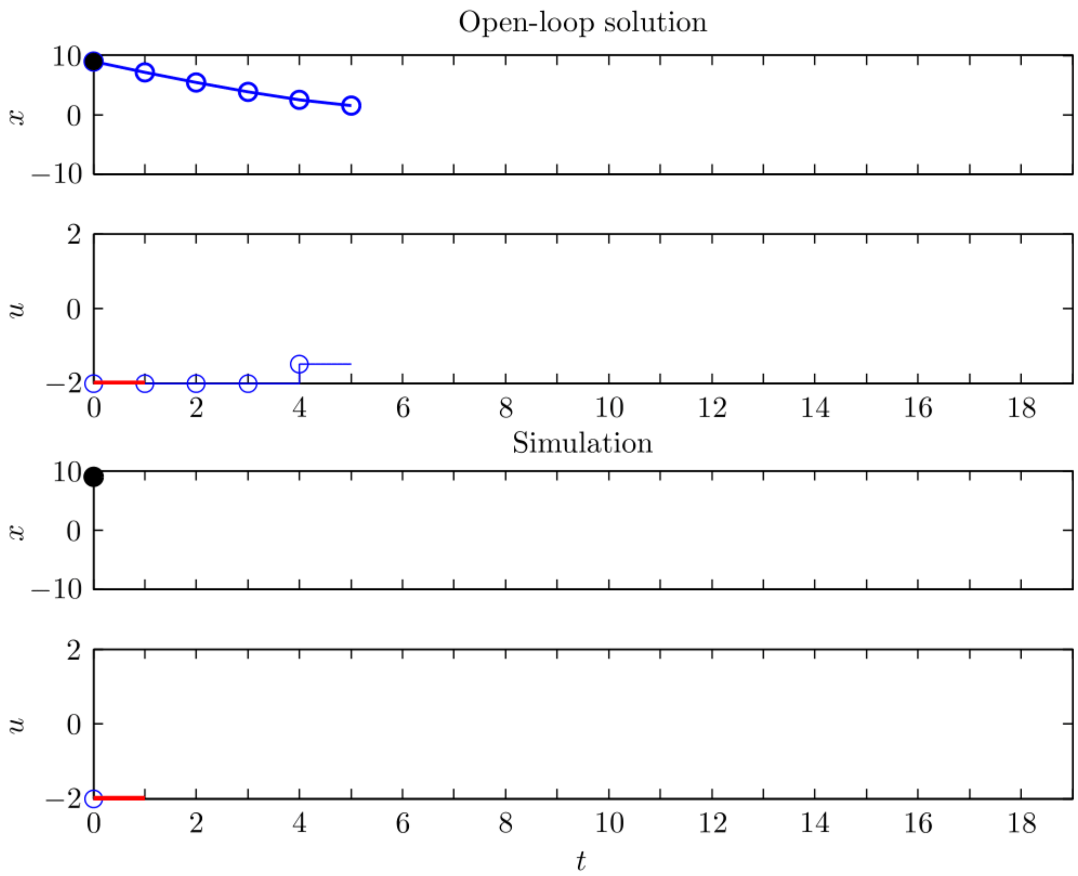
NTNU | Norwegian University of Science and Technology



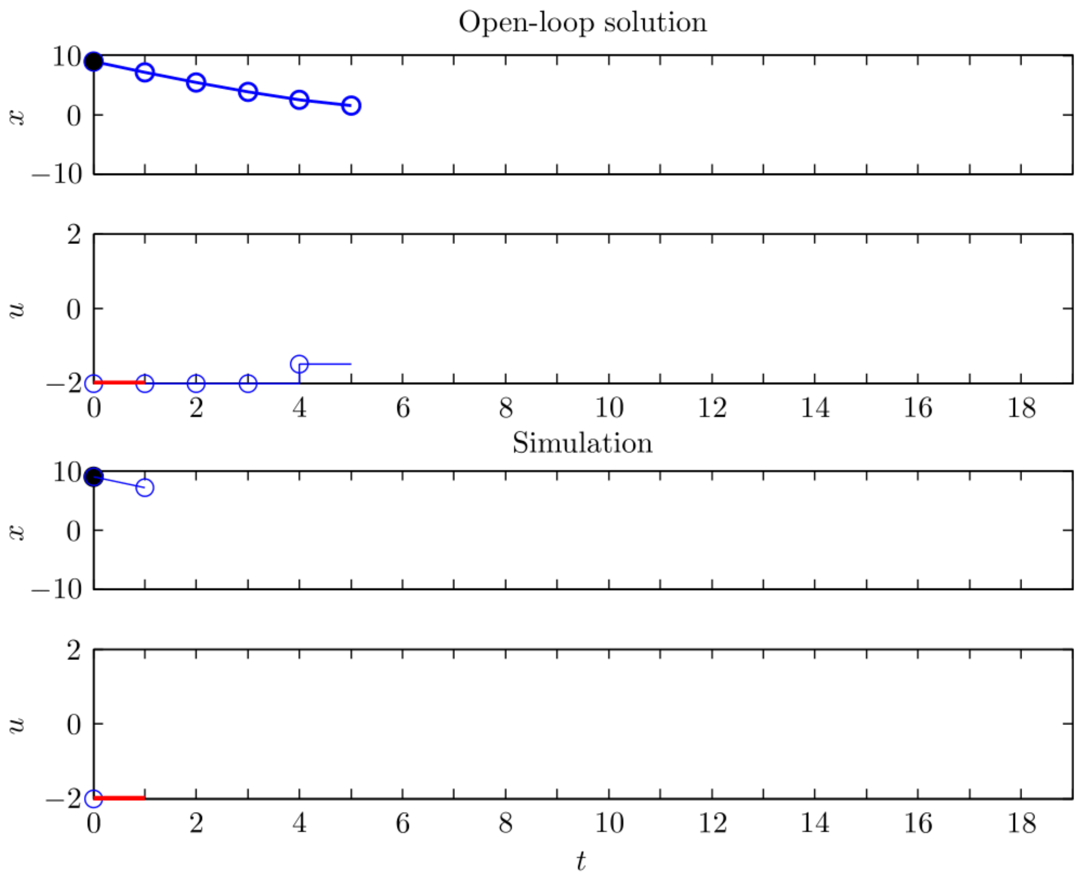
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t



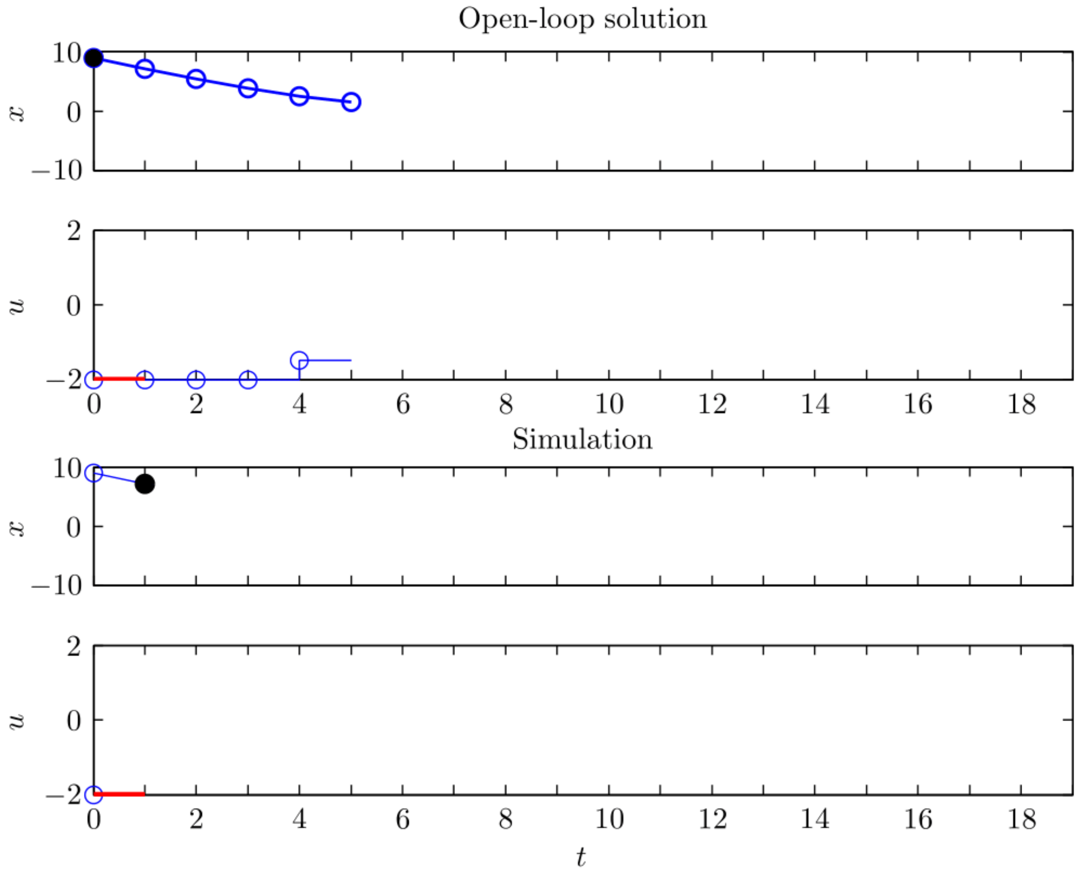
Open-loop solution
Simulation
x
u
t
10
0
−10
2
0
−2
0
2
4
6
8
10
12
14
16
18

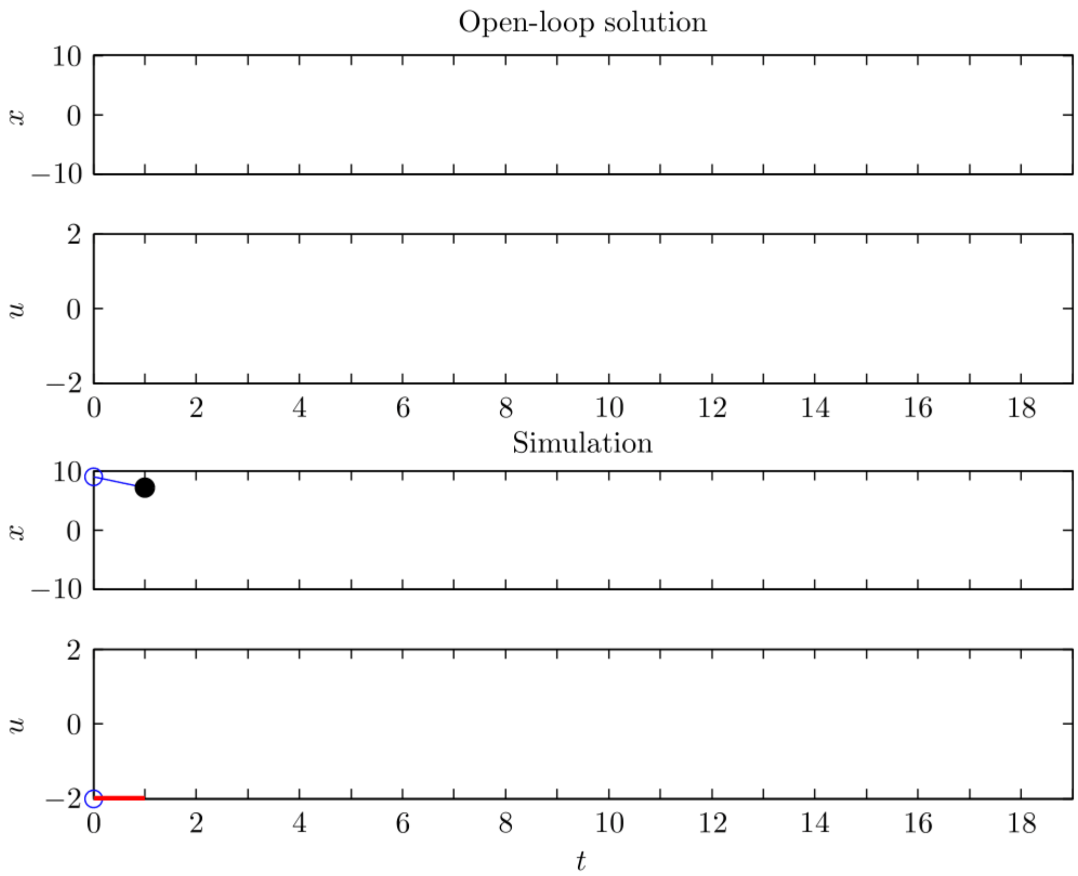


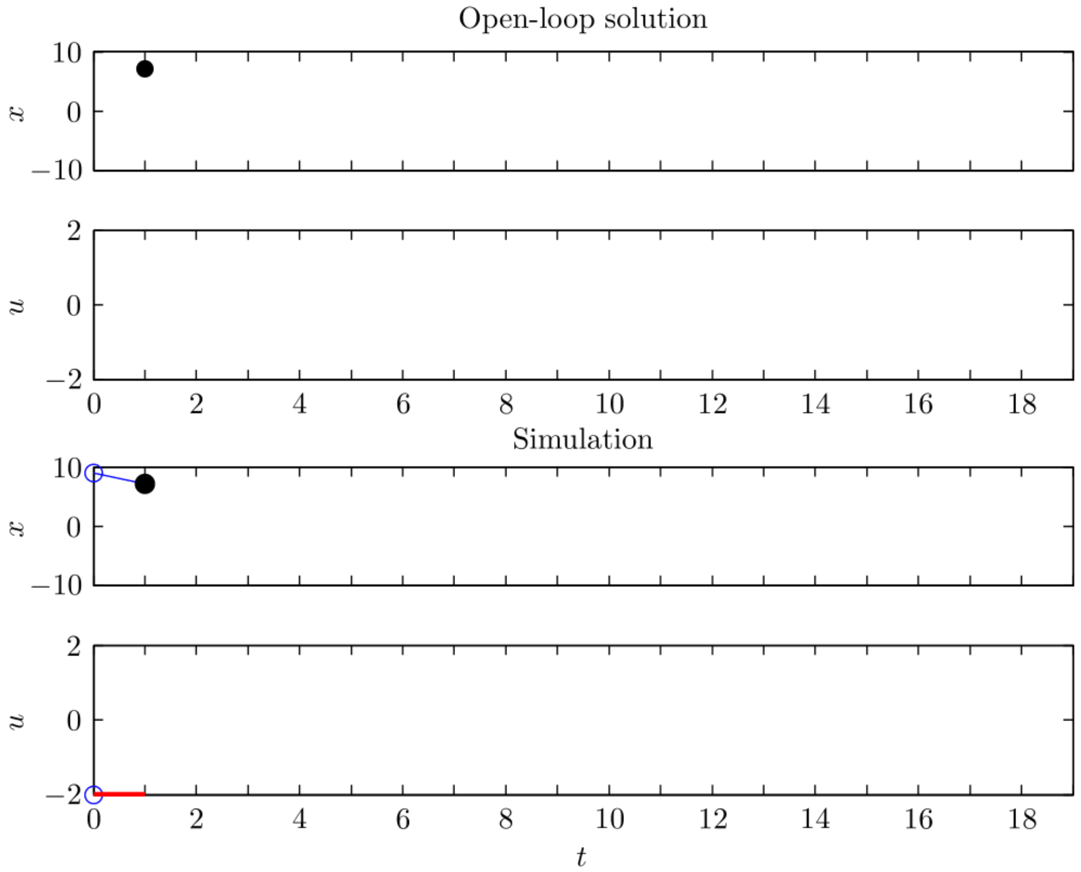
Open-loop solution
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
Simulation
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
t

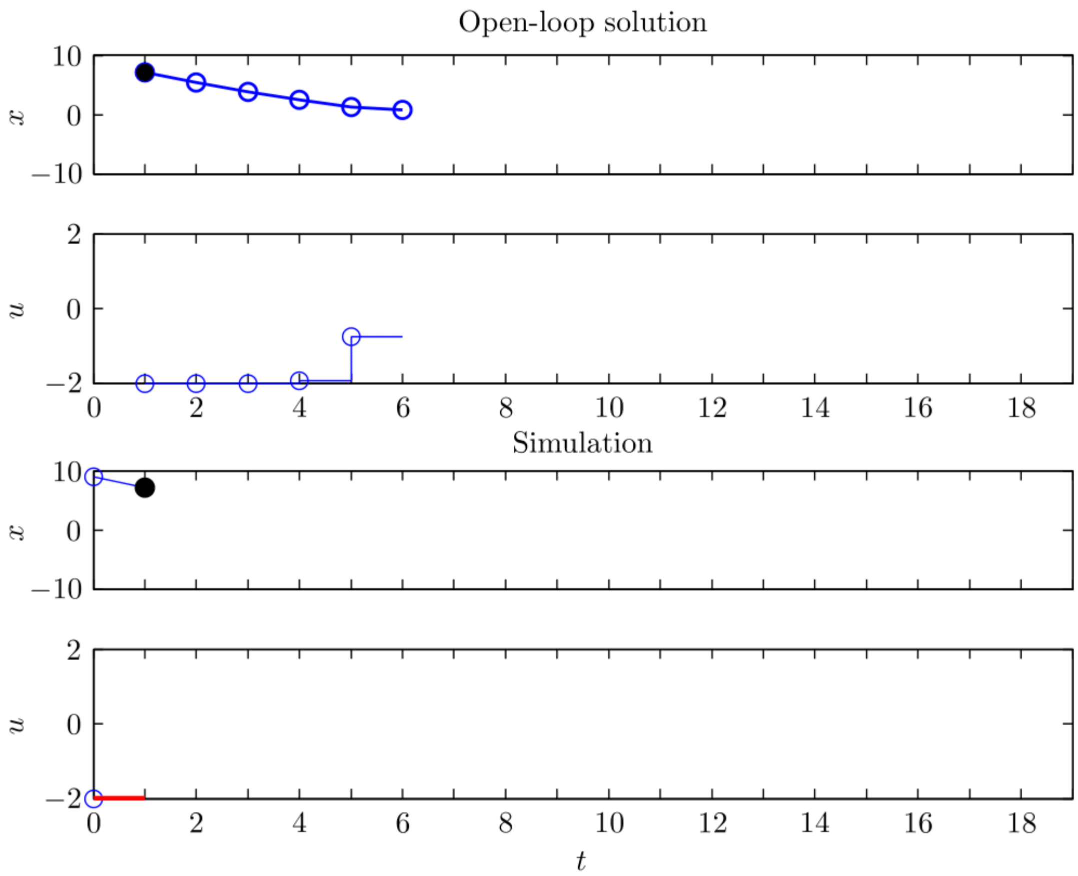


Open-loop solution
x
10
0
−10
u
2
0
−2
0 2 4 6 8 10 12 14 16 18
Simulation
x
10
0
−10
u
2
0
−2
0 2 4 6 8 10 12 14 16 18
t

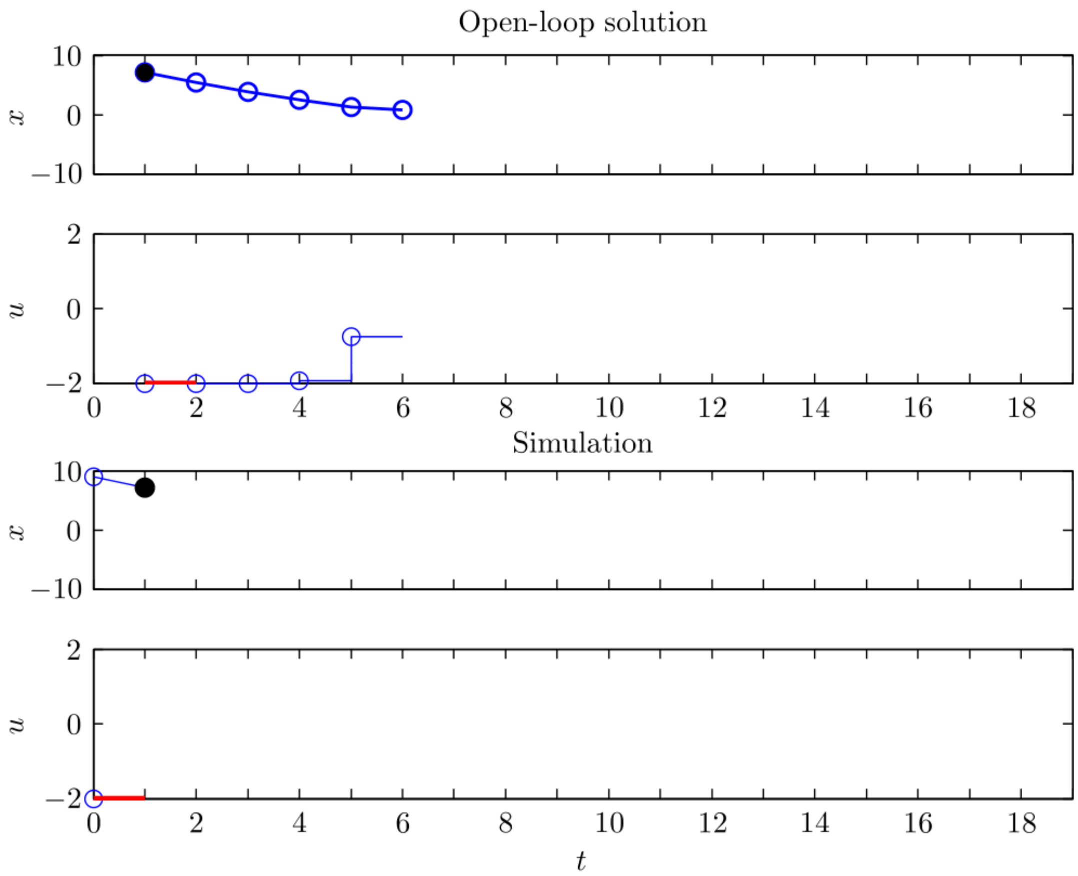

Open-loop solution
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
Simulation
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
t


Open-loop solution
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
Simulation
10
0
−10
x
2
0
−2
u
0 2 4 6 8 10 12 14 16 18
t

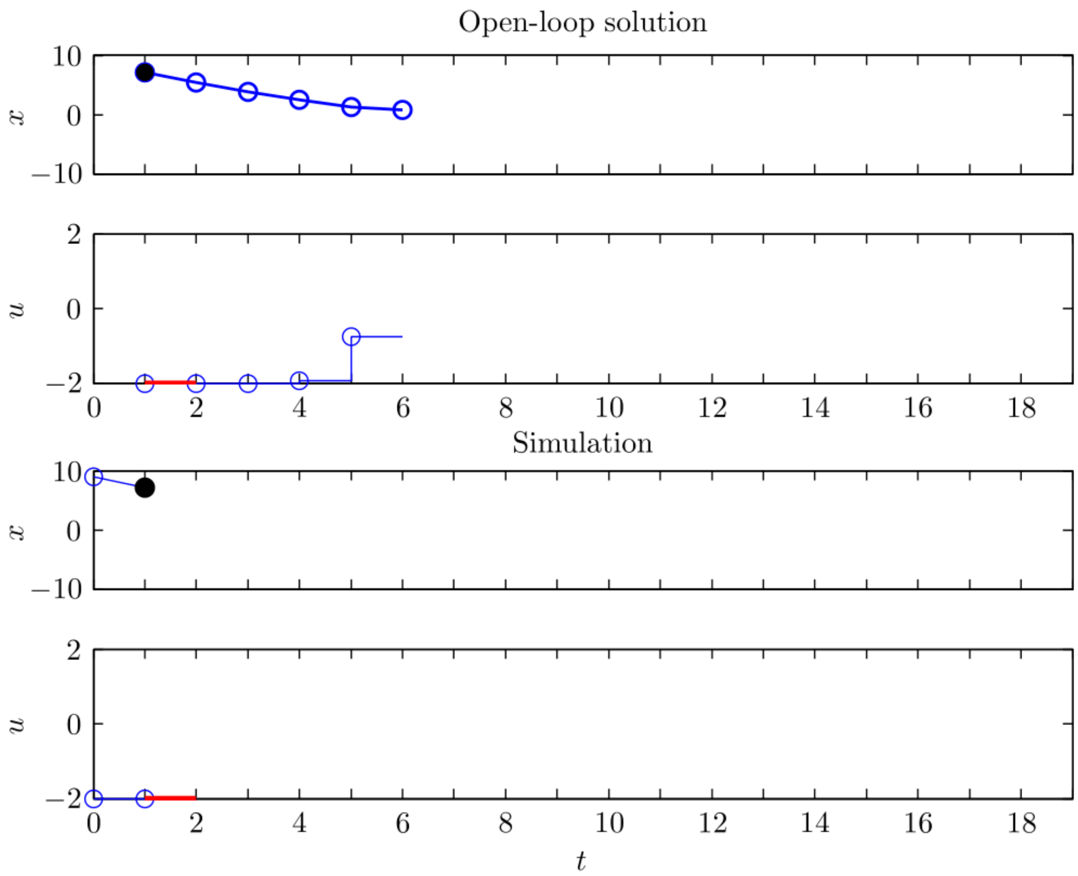

Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t



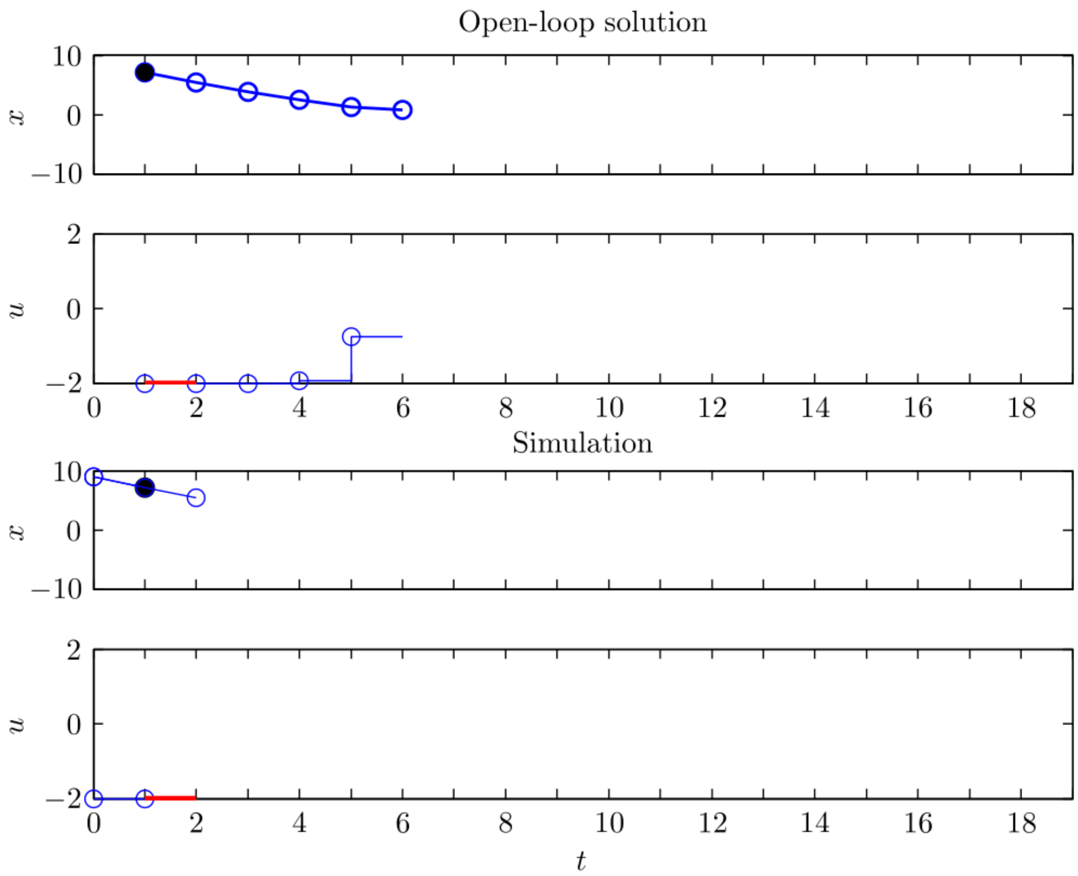
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t



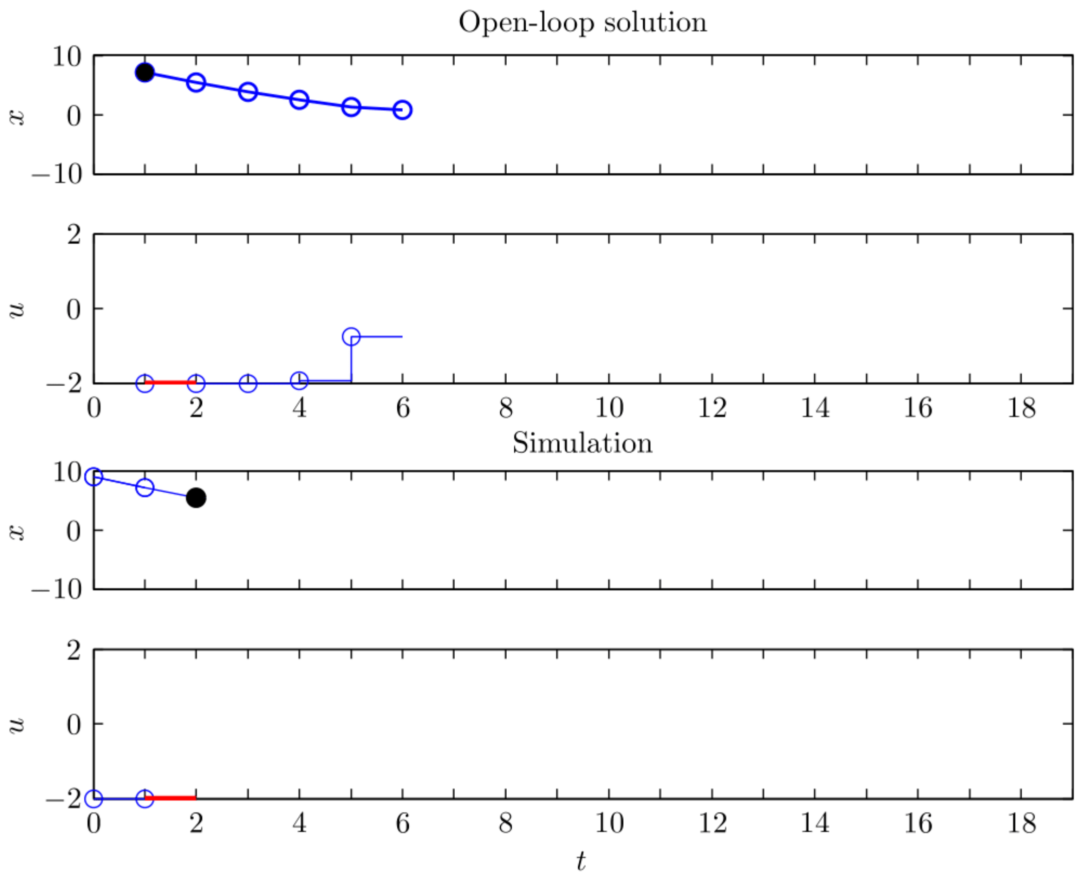
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t

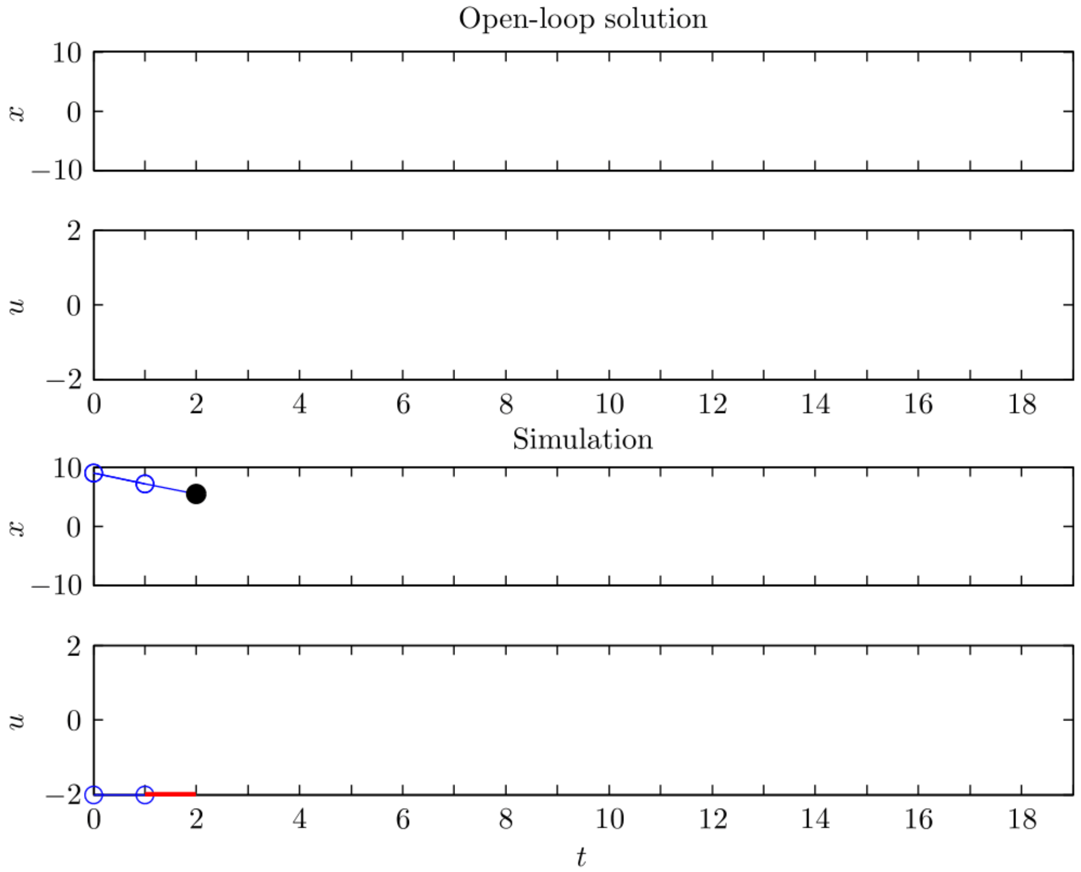


Open-loop solution
x
u
Simulation
x
u
t

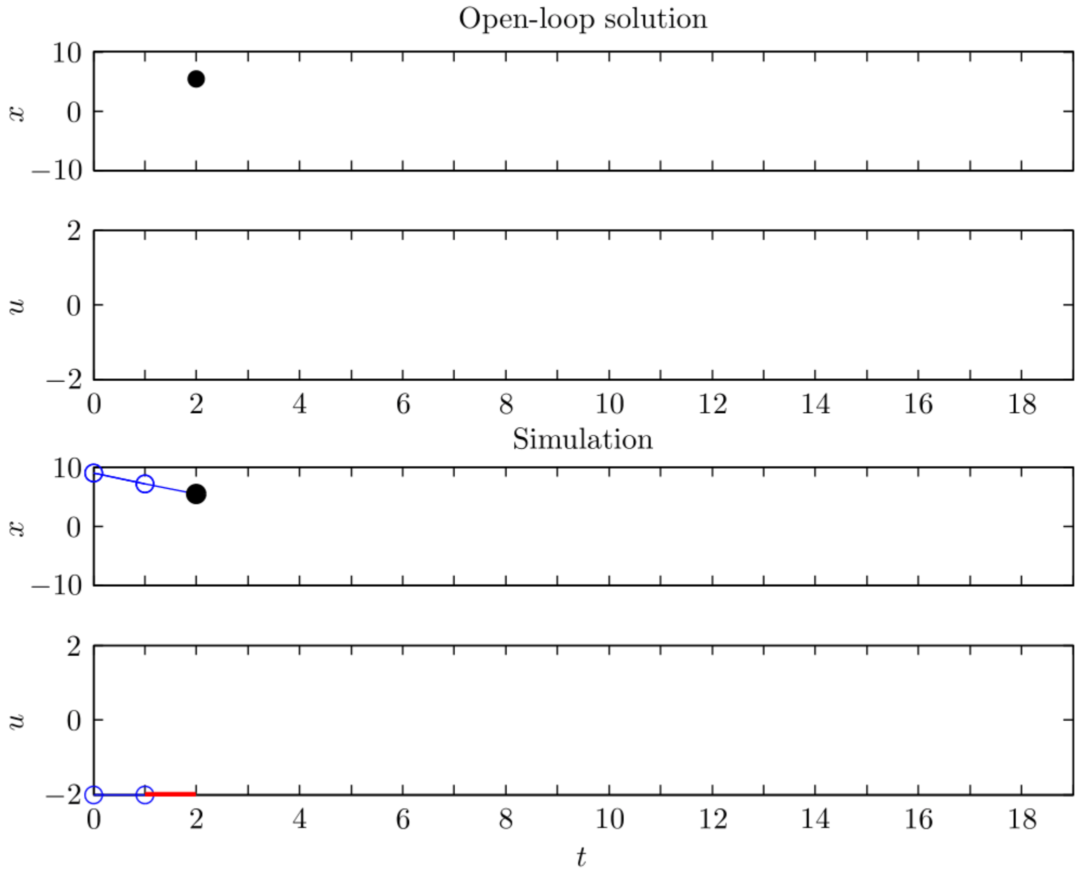


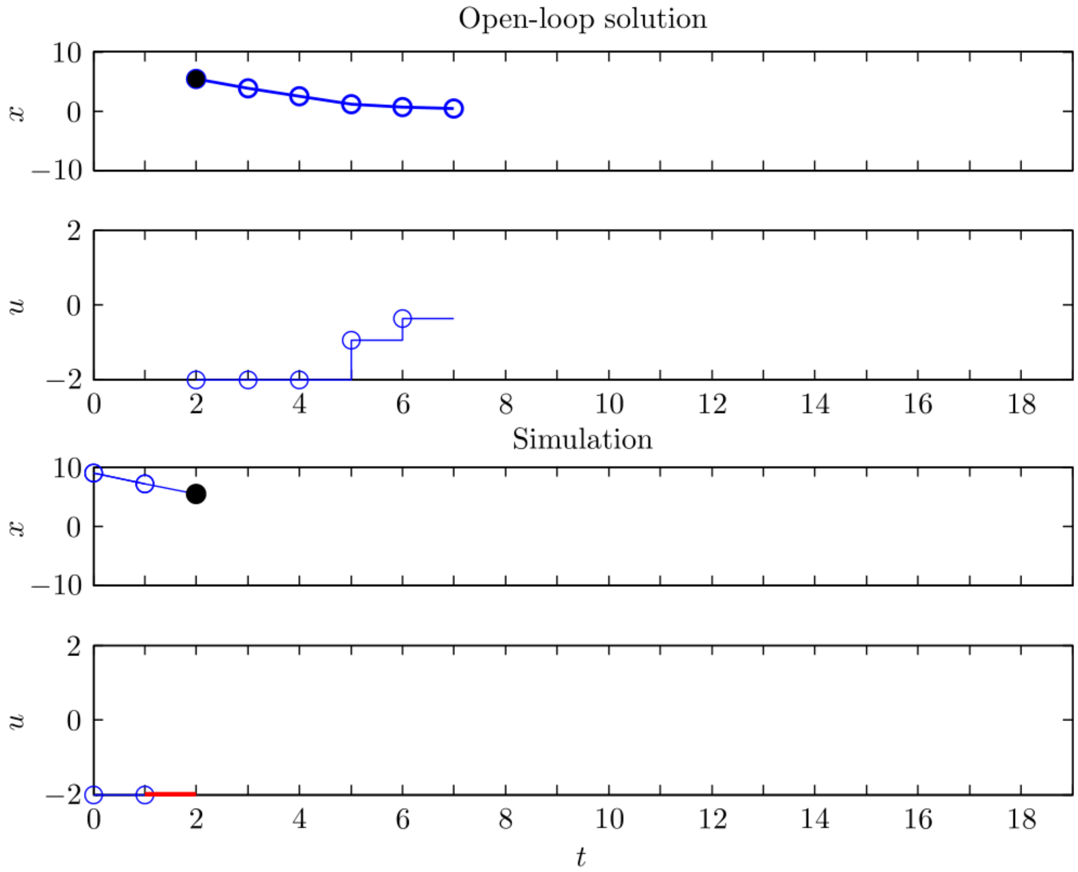
NTNU | Norwegian University of Science and Technology


Open-loop solution
Simulation
x
u
t

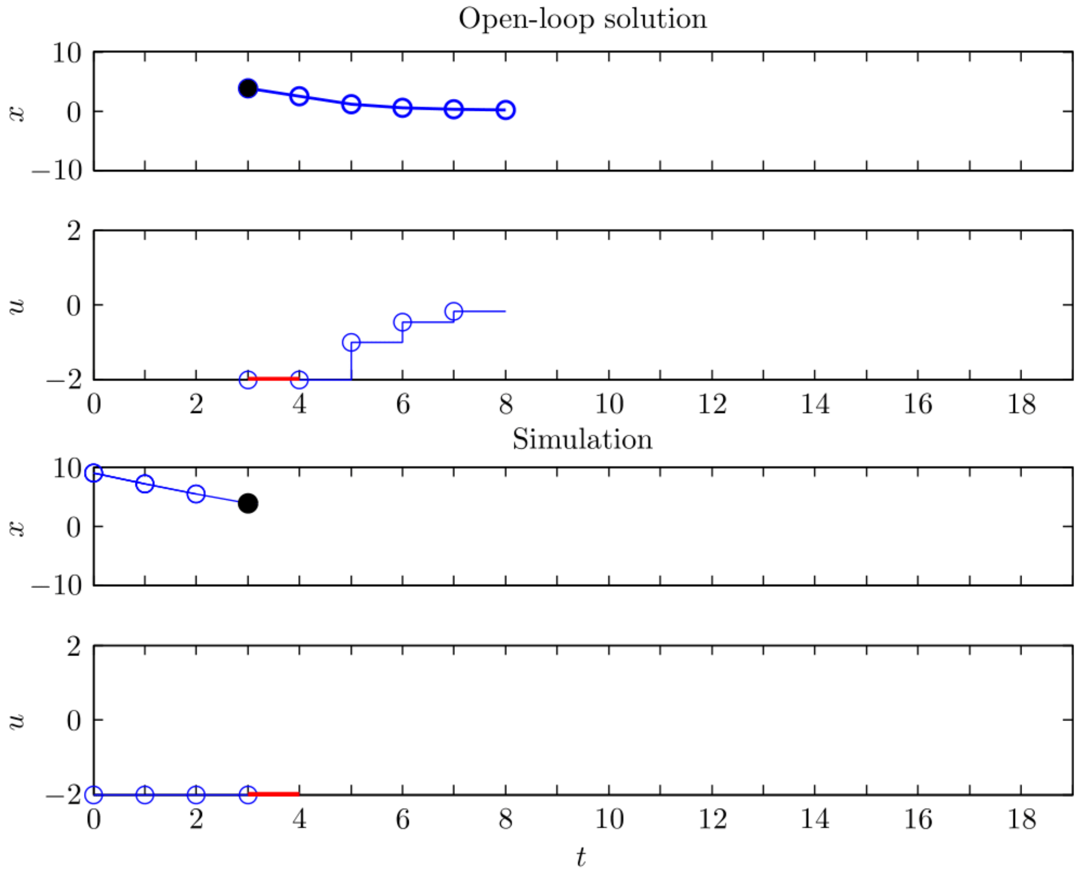


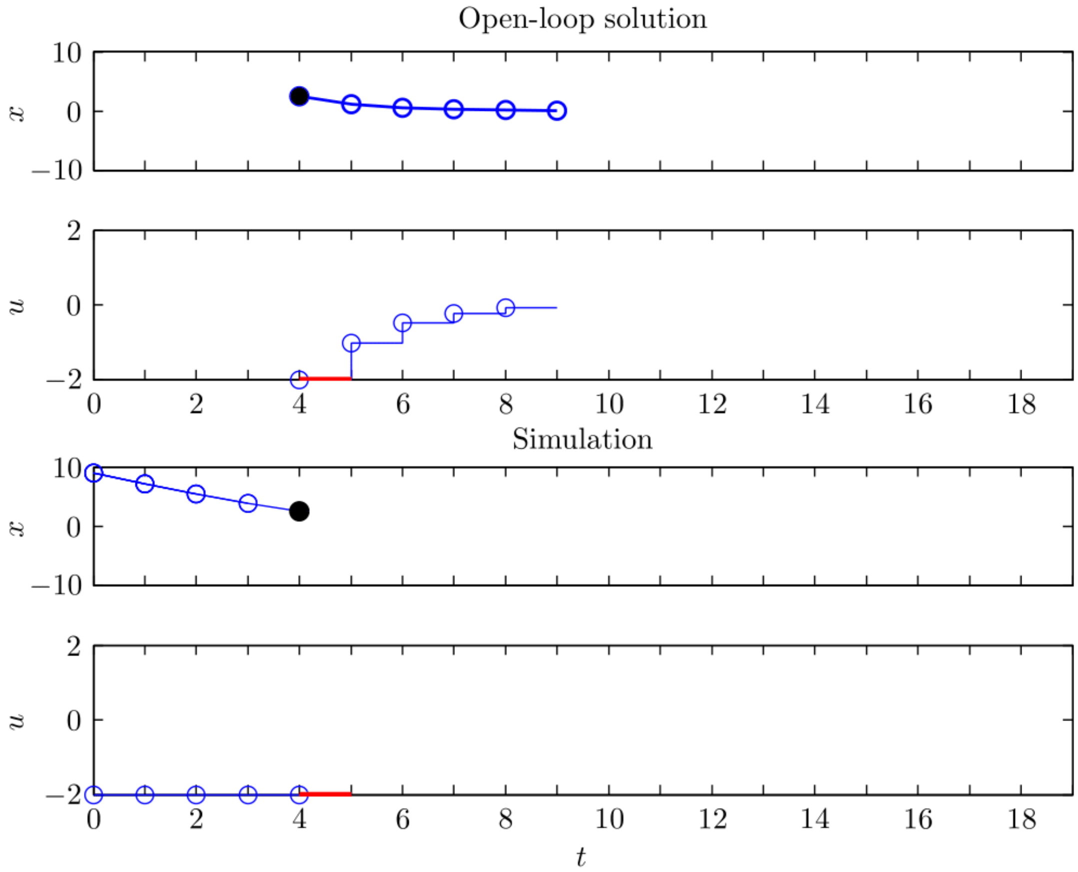
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t

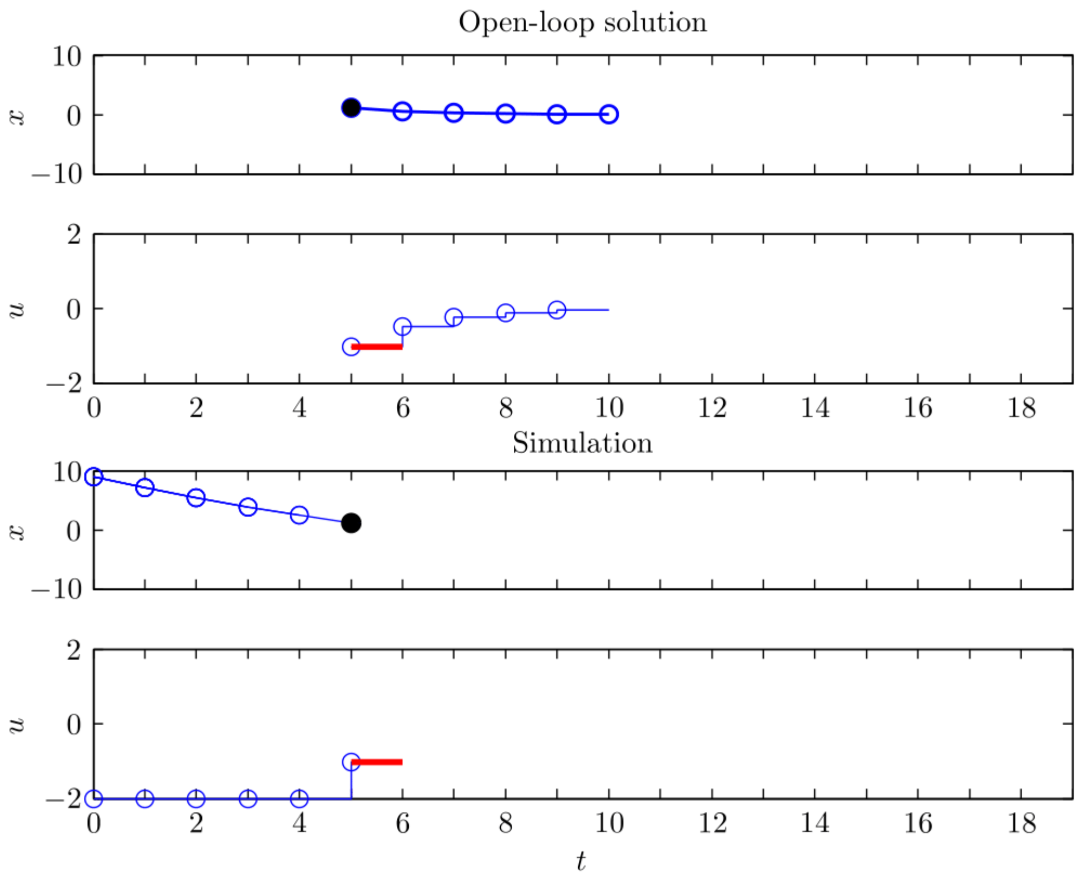

Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t



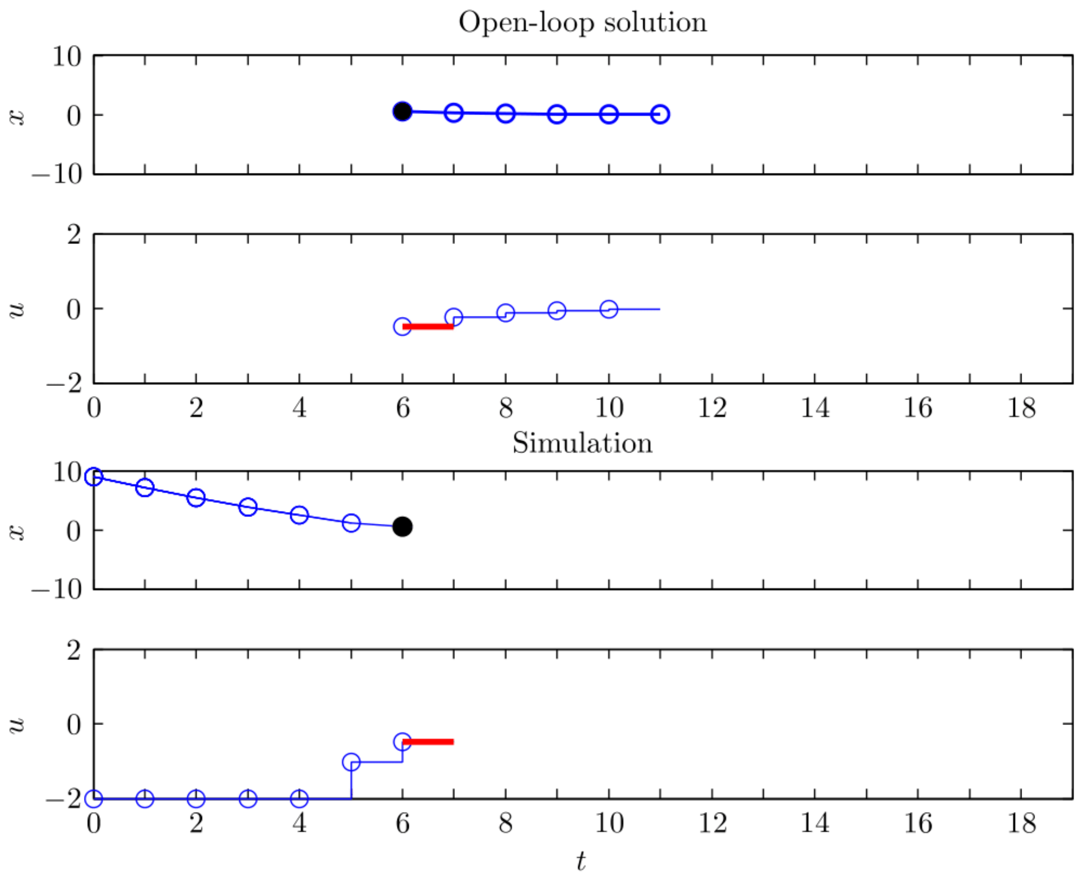
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t


Open-loop solution
Simulation
x
u
t
10
0
−10
2
−2
0 2 4 6 8 10 12 14 16 18

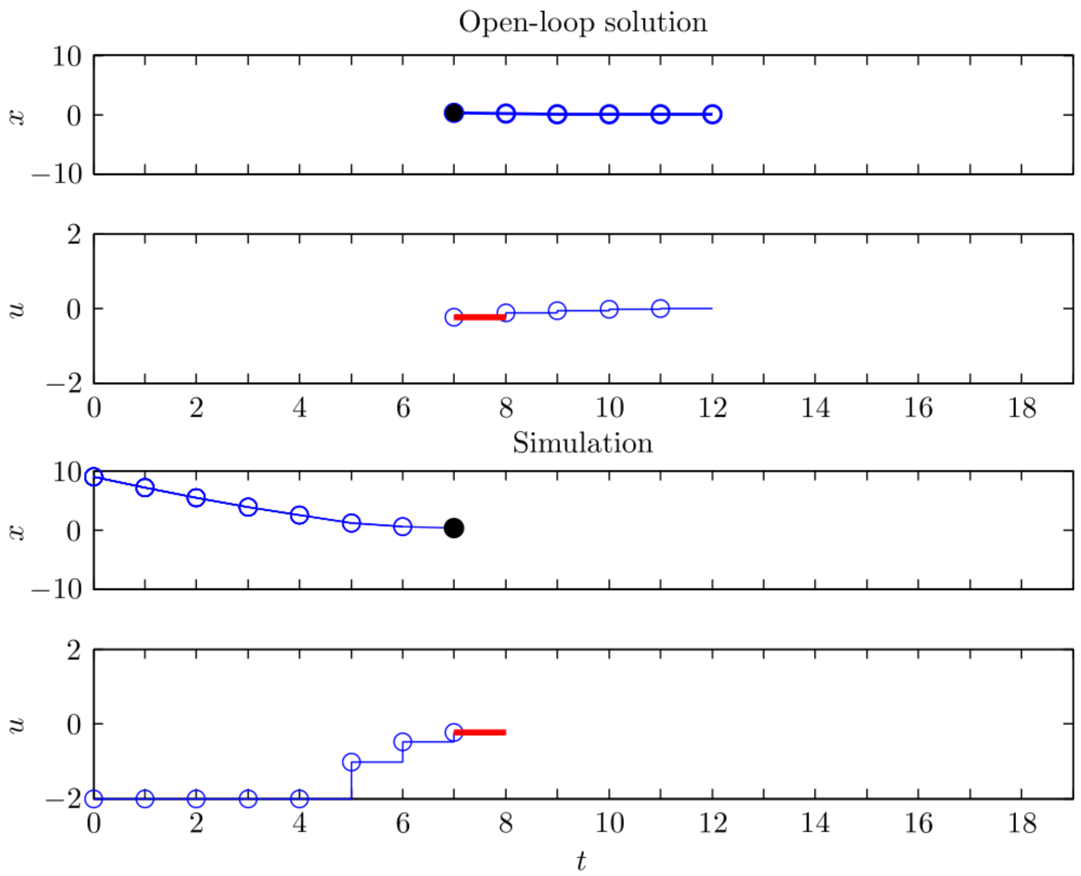

Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t

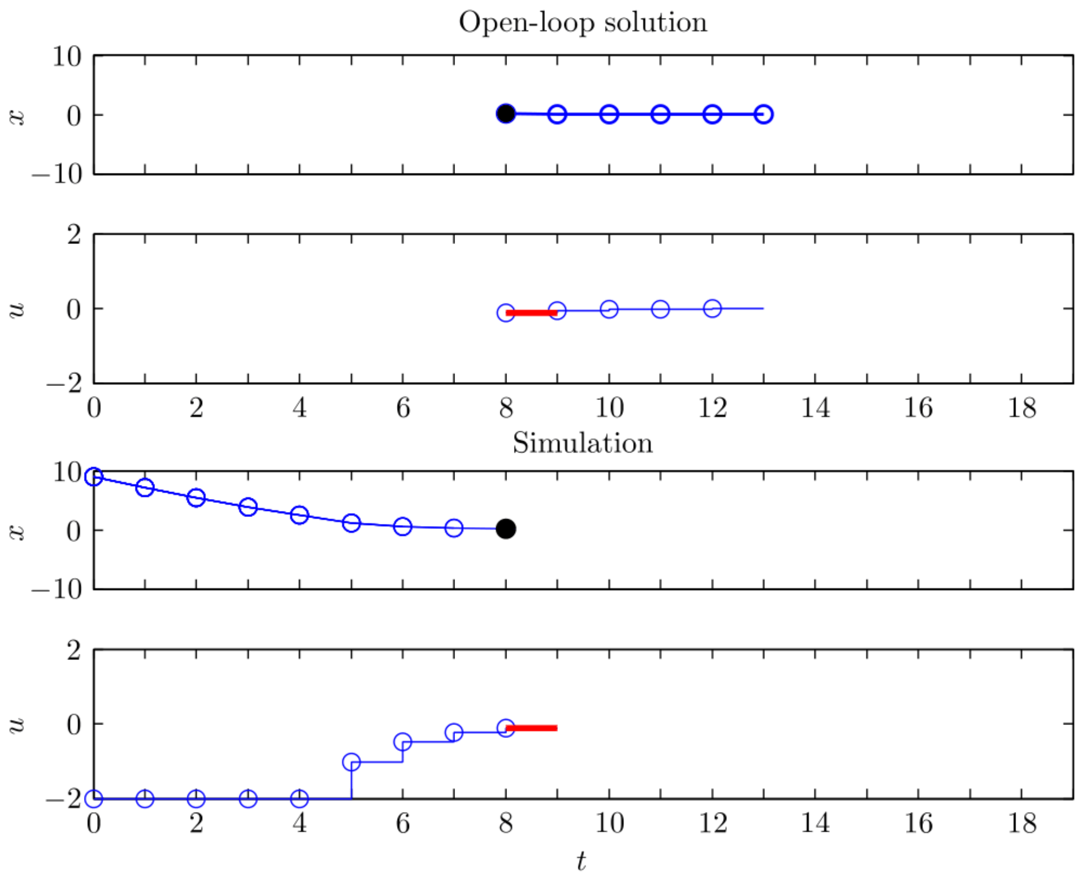


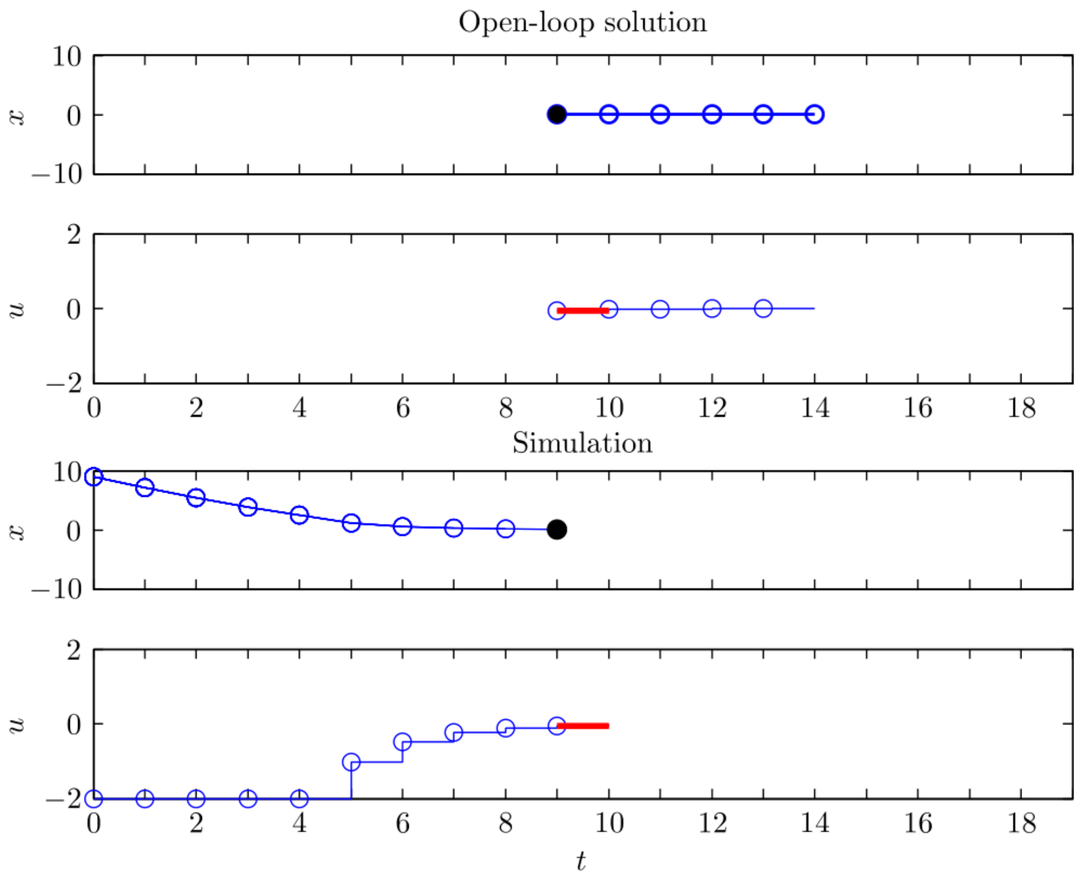
Open-loop solution
x
u
Simulation
x
u
t
10
0
−10
2
0
−2
0
2
4
6
8
10
12
14
16
18



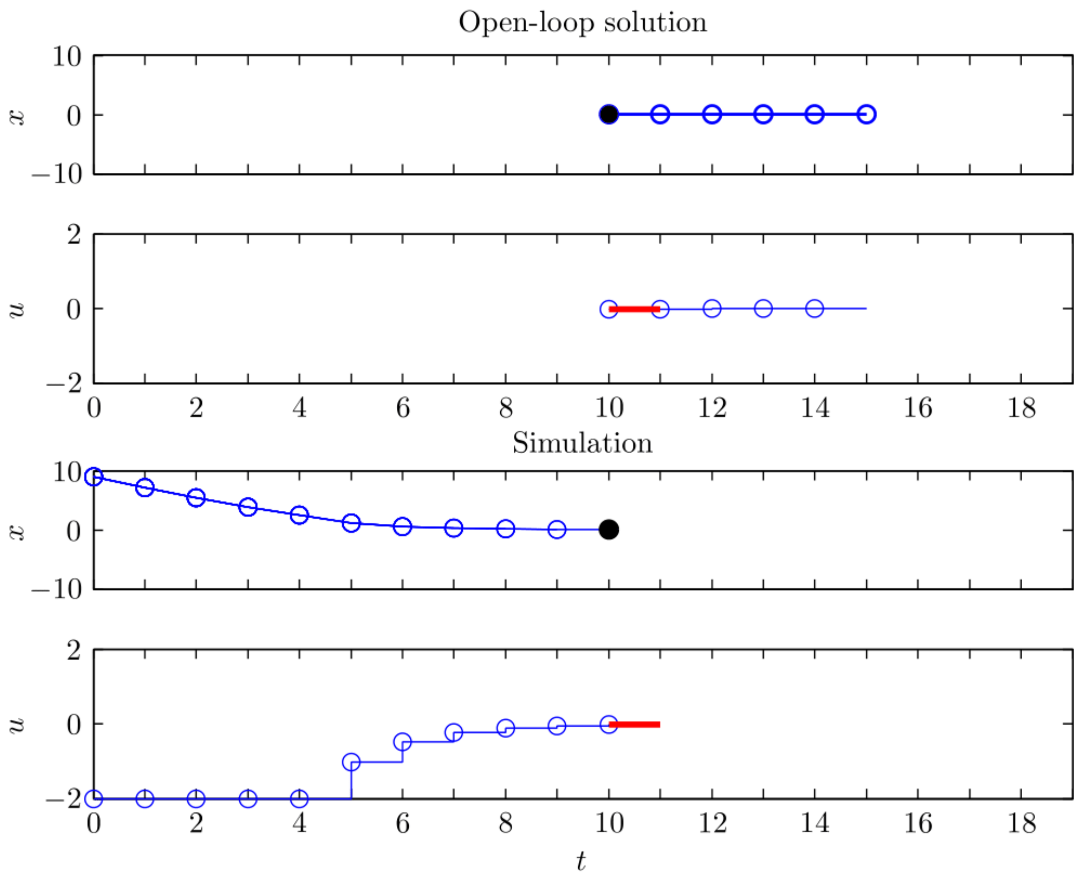
Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t

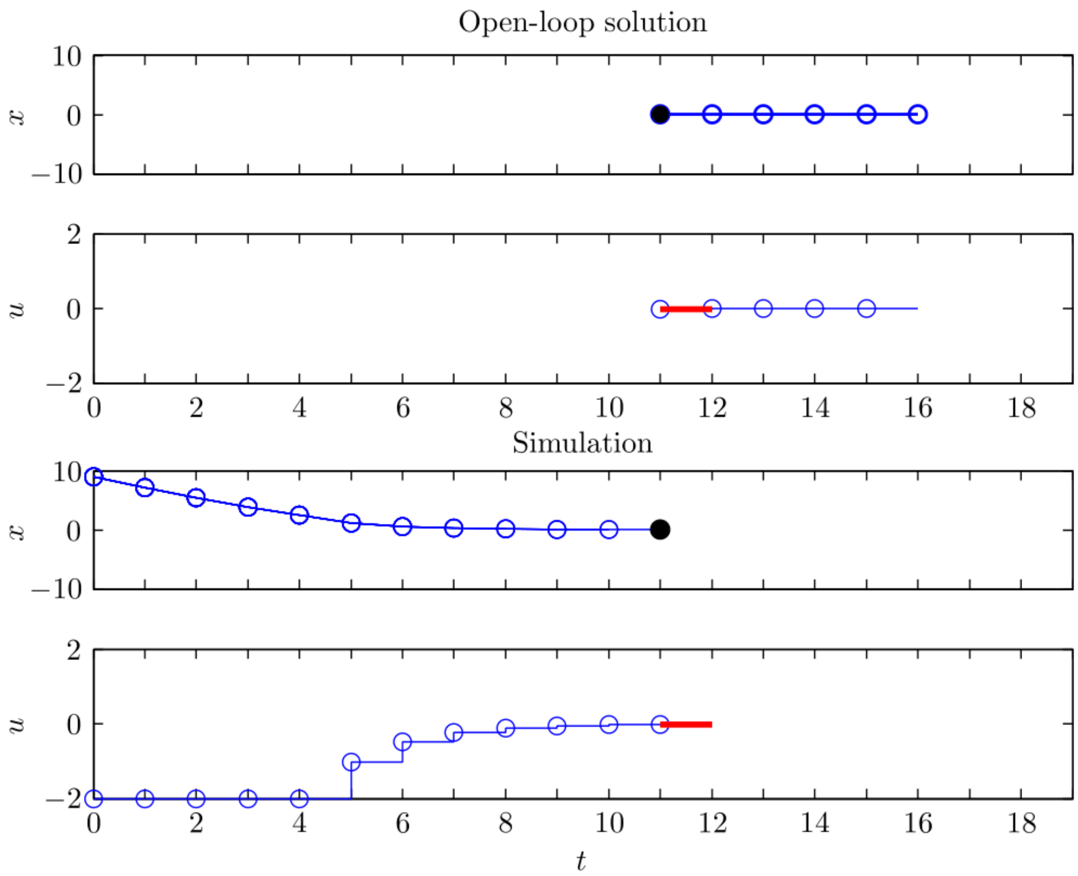

Open-loop solution
Simulation
x
u
t
10
0
−10
2
−2
0
2
4
6
8
10
12
14
16
18

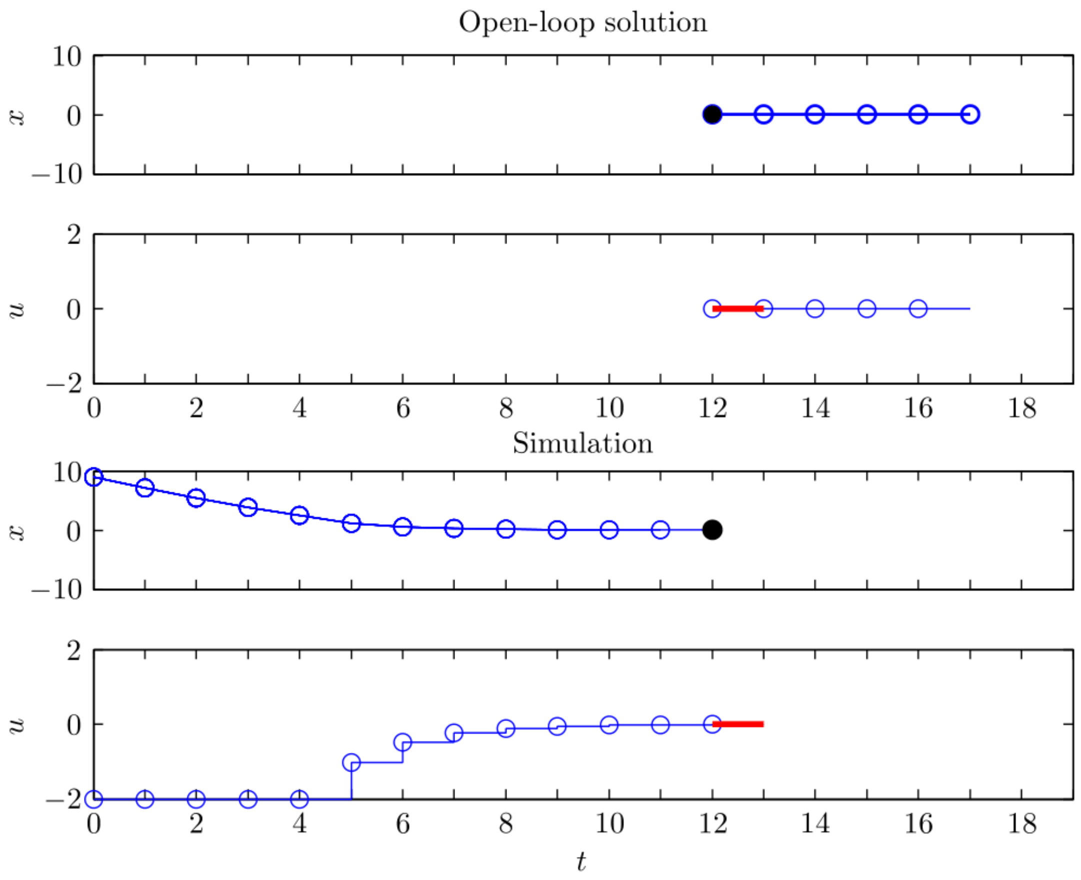

Open-loop solution
x
u
Simulation
x
u
t

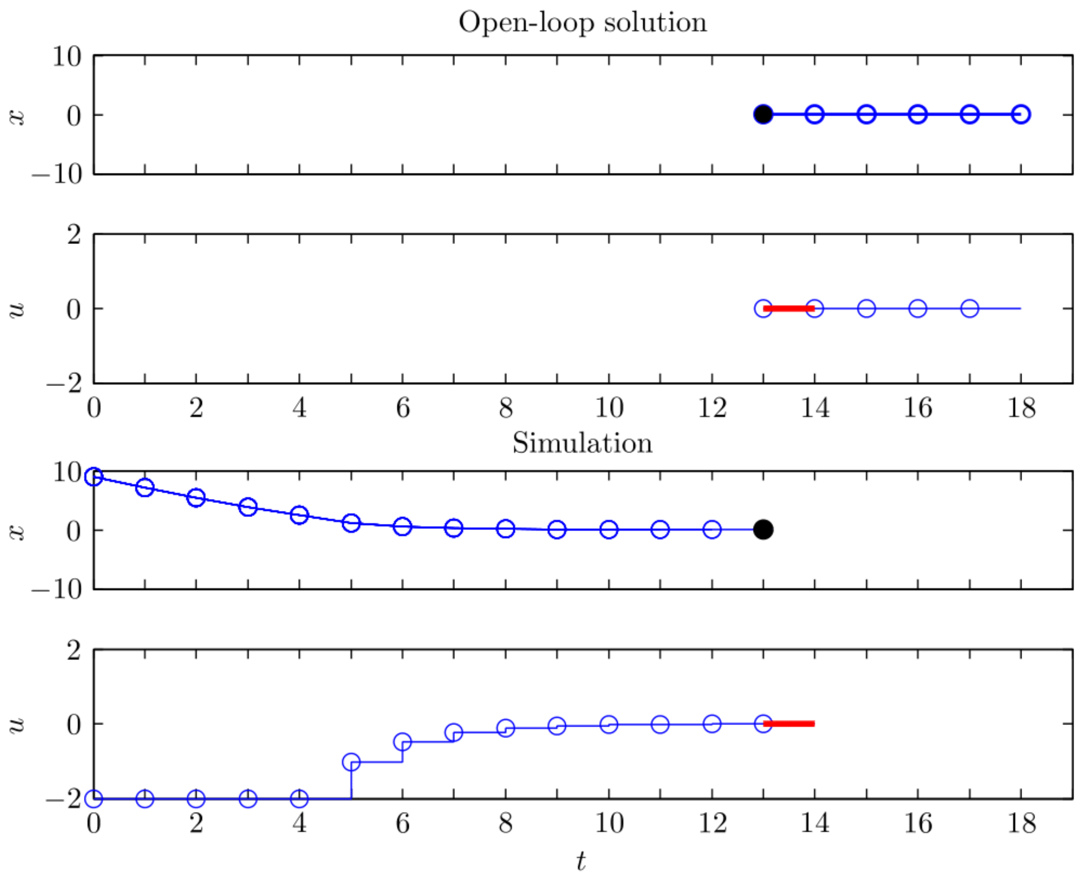


Open-loop solution
Simulation
x
u
t

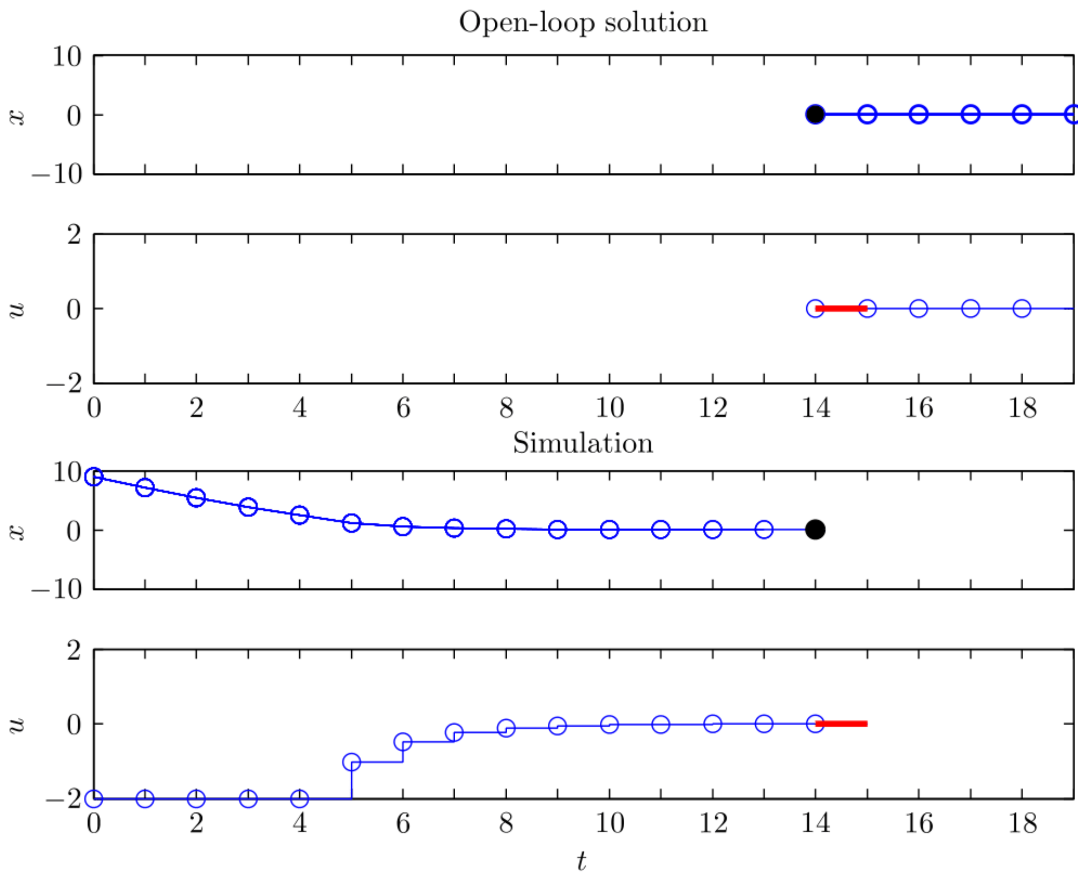

Open-loop solution
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t


Open-loop solution
x
u
Simulation
t
10
0
−10
2
−2
0 2 4 6 8 10 12 14 16 18


Open-loop solution
Simulation
10
0
−10
x
2
0
−2
u
0
2
4
6
8
10
12
14
16
18
t



Norwegian University of Science and Technology


Open-loop solution
x
u
Simulation
x
u
t
10
0
−10
2
0
−2
0 2 4 6 8 10 12 14 16 18

Open-loop optimization with linear state-space model

QP

$$\min_{z \in \mathbb{R}^n} f(z) = \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q_{t+1} x_{t+1} + d_{x,t+1} x_{t+1} + \frac{1}{2} u_t^\top R_t u_t + d_{u,t} u_t + \frac{1}{2} \Delta u_t^\top S \Delta u_t$$

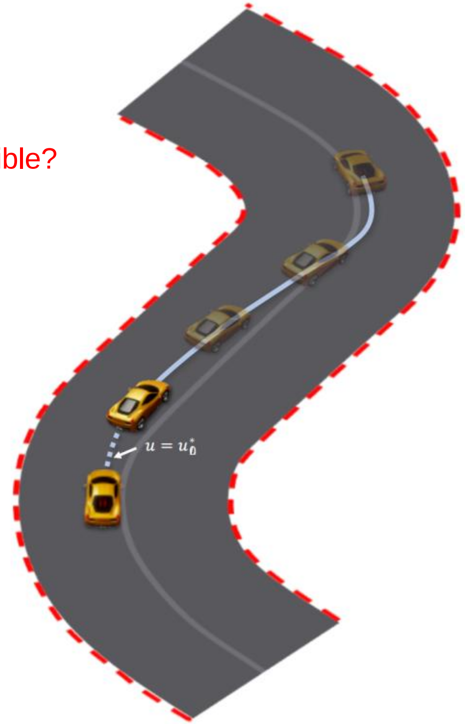
subject to

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t, \quad t = \{0, \dots, N-1\} \\ x^{\text{low}} &\leq x_t \leq x^{\text{high}}, \quad t = \{1, \dots, N\} \\ u^{\text{low}} &\leq u_t \leq u^{\text{high}}, \quad t = \{0, \dots, N-1\} \\ -\Delta u^{\text{high}} &\leq \Delta u_t \leq \Delta u^{\text{high}}, \quad t = \{0, \dots, N-1\} \end{aligned}$$

Is this always possible?

where

$$\begin{aligned} x_0 &\text{ and } u_{-1} \text{ is given} \\ \Delta u_t &:= u_t - u_{t-1} \\ z^\top &:= (u_0^\top, x_1^\top, \dots, u_{N-1}^\top, x_N^\top) \\ n &= N \cdot (n_x + n_u) \\ Q_t &\succeq 0 \quad t = \{1, \dots, N\} \\ R_t &\succeq 0 \quad t = \{0, \dots, N-1\} \end{aligned}$$

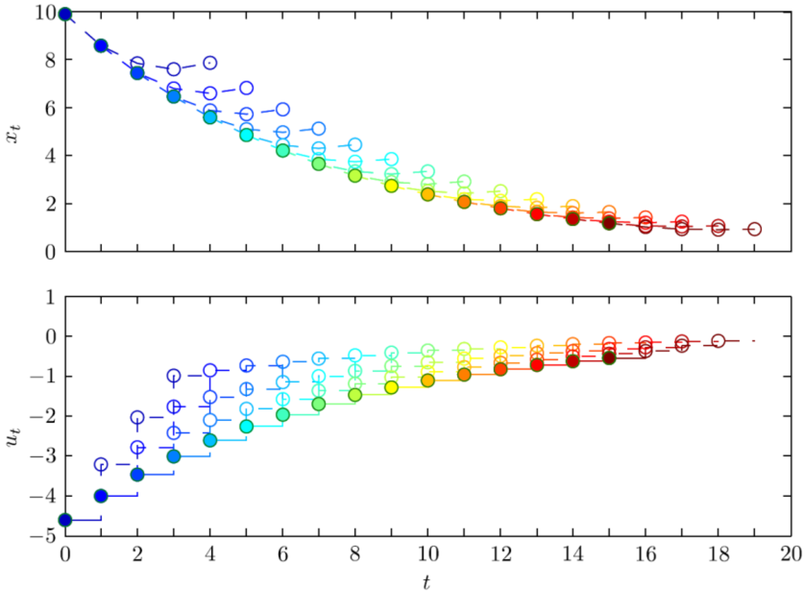


The **feasibility** problem: Inequality constraints on states may imply that for some x_0, there are no solutions to the MPC QP

Open-loop vs closed-loop trajectories

$$\min \sum_{t=0}^{4} x_{t+1}^2 + 4\,u_t^2$$

$$\text{s.t.} \qquad x_{t+1} = 1.2x_t + 0.5u_t, \quad t = 0, \ldots, 4$$



- Closed-loop trajectories different from open-loop (optimized) trajectories!
- It is the closed-loop trajectories that must analyzed for stability

Example: Is MPC alway stable?

Design MPC for $x_{t+1} = 1.2x_t + u_t$, no constraints, *N = 2*

Example: Is MPC alway stable?

Design MPC for $x_{t+1} = 1.2x_t + u_t$, no constraints, *N* = *2*

MPC optimality implies stability?

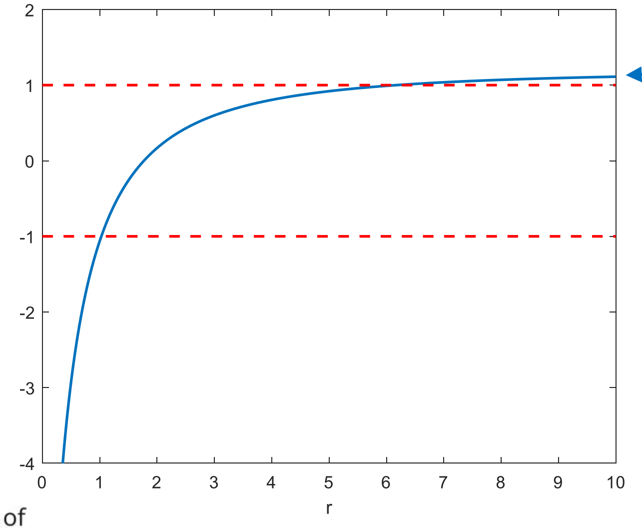
$$\min \sum_{t=0}^{1} x_{t+1}^2 + r\, u_t^2$$
$$\text{s.t.} \quad x_{t+1} = 1.2x_t + u_t, \quad t = 0, 1$$

MPC solution

$$u_t = -\frac{1.2 + 2.64r}{1 + 3.2r + r^2} x_t$$

MPC closed loop

$$x_{t+1} = \underbrace{\left(1.2 - \frac{1.2 + 2.64r}{1 + 3.2r + r^2}\right)}\, x_t$$



MPC and stability

Nominal vs robust stability

- “Nominal stability”: Stability when optimization model = plant model
  - No “model-plant mismatch”, no disturbances
- “Robust stability”: Stability when optimization model ≠ plant model
  - “Model-plant mismatch” and/or disturbances (more difficult to analyze, not part of this course)

Requirements for nominal stability:

- Stabilizability ((A,B) stabilizable)
- Detectability ((A,D) detectable)
  - D is a matrix such that $Q = D^T D$ (that is, “D is matrix square root of Q”)
  - Detectability: No modes can grow to infinity without being “visible” through Q

- But more is needed to guarantee stability…

How to achieve nominal stability?

$$\begin{aligned}
\min_{z} \quad & \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1 \\
& x^{\text{low}} \leq x_t \leq x^{\text{high}}, \quad t = 1, \dots, N \\
& u^{\text{low}} \leq u_t \leq u^{\text{high}}, \quad t = 0, \dots, N-1
\end{aligned}$$

Three different theoretical solutions:

1. Choose prediction horizon equal to infinity ($N = \infty$)
   - Usually not possible (unless no constraints: LQR)

How to achieve nominal stability?

$$
\begin{aligned}
\min_{z} \quad & \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1 \\
& x^{\text{low}} \leq x_t \leq x^{\text{high}}, \quad t = 1, \dots, N \\
& u^{\text{low}} \leq u_t \leq u^{\text{high}}, \quad t = 0, \dots, N-1
\end{aligned}
$$

Three different theoretical solutions:

1. Choose prediction horizon equal to infinity ($N = \infty$)
   - Usually not possible (unless no constraints: LQR)
2. For given N, design Q and R such that MPC is stable (cf. example)
   - Difficult in general! And usually we want to design (“tune”) Q and R for performance, not for stability

How to achieve nominal stability?

$$
\begin{aligned}
\min_{z} \quad & \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1 \\
& x^{\text{low}} \leq x_t \leq x^{\text{high}}, \quad t = 1, \dots, N \\
& u^{\text{low}} \leq u_t \leq u^{\text{high}}, \quad t = 0, \dots, N-1
\end{aligned}
$$

Three different theoretical solutions:

1. Choose prediction horizon equal to infinity ($N = \infty$)
   - Usually not possible (unless no constraints: LQR)
2. For given N, design Q and R such that MPC is stable (cf. example)
   - Difficult in general! And usually we want to design ("tune") Q and R for performance
3. Change the optimization problem: Terminal cost + terminal constraint
   - Choose terminal cost + terminal constraint such that cost of new problem is a feasible upper bound on cost of infinite horizon problem
   - General theory for finding such terminal cost & terminal constraint exist, not difficult, but may be somewhat impractical for "real" problems

Terminal cost

$$
\begin{aligned}
\min_{z} \quad & \sum_{t=0}^{N-1} \left(\frac{1}{2} x_t^\top Q x_t + \frac{1}{2} u_t^\top R u_t \right) + \frac{1}{2} x_N^\top P x_N \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1 \\
& x^{\text{low}} \leq x_t \leq x^{\text{high}}, \quad t = 1, \dots, N \\
& u^{\text{low}} \leq u_t \leq u^{\text{high}}, \quad t = 0, \dots, N-1 \\
& x_N \in \mathcal{S}
\end{aligned}
$$

Terminal constraint

How to achieve nominal stability?

$$\min_{z} \sum_{t=0}^{N-1} \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t$$
$$\text{s.t.} \quad x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1$$
$$x^{\text{low}} \le x_t \le x^{\text{high}}, \quad t = 1, \dots, N$$
$$u^{\text{low}} \le u_t \le u^{\text{high}}, \quad t = 0, \dots, N-1$$

Three different theoretical solutions:

1. Choose prediction horizon equal to infinity (*N* = ∞)
   - Usually not possible (unless no constraints: LQR)
2. For given *N*, design *Q* and *R* such that MPC is stable (cf. example)
   - Difficult in general! And usually we want to design ("tune") *Q* and *R* for performance
3. Change the optimization problem: Terminal cost + terminal constraint
   - Choose terminal cost + terminal constraint such that cost of new problem is a feasible upper bound on cost of infinite horizon problem
   - General theory for finding such terminal cost & terminal constraint exist, not difficult, but may be somewhat impractical for "real" problems

Terminal cost

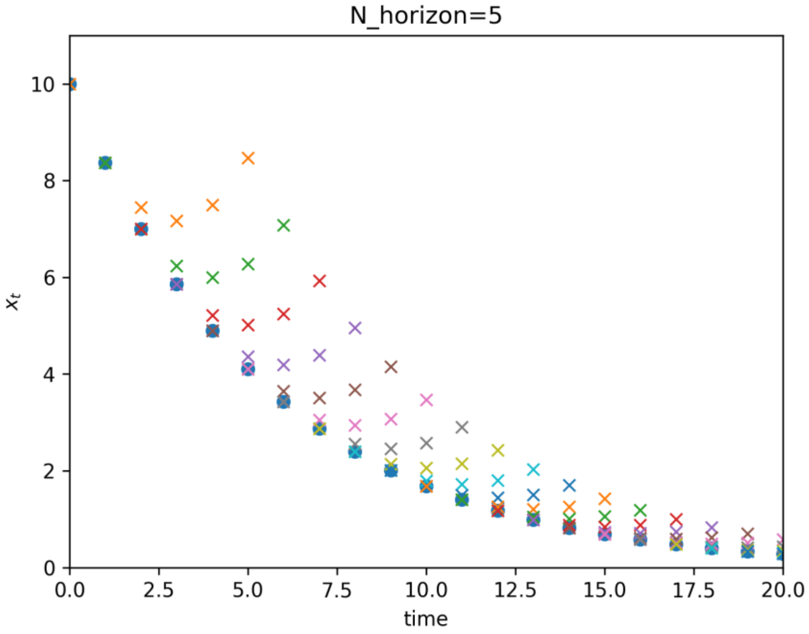
$$\min_{z} \sum_{t=0}^{N-1} \left(\frac{1}{2} x_t^\top Q x_t + \frac{1}{2} u_t^\top R u_t \right) + \frac{1}{2} x_N^\top P x_N$$
$$\text{s.t.} \quad x_{t+1} = A x_t + B u_t, \quad t = 0, 1, \dots, N-1$$
$$x^{\text{low}} \le x_t \le x^{\text{high}}, \quad t = 1, \dots, N$$
$$u^{\text{low}} \le u_t \le u^{\text{high}}, \quad t = 0, \dots, N-1$$
$$x_N \in \mathcal{S}$$

Terminal constraint

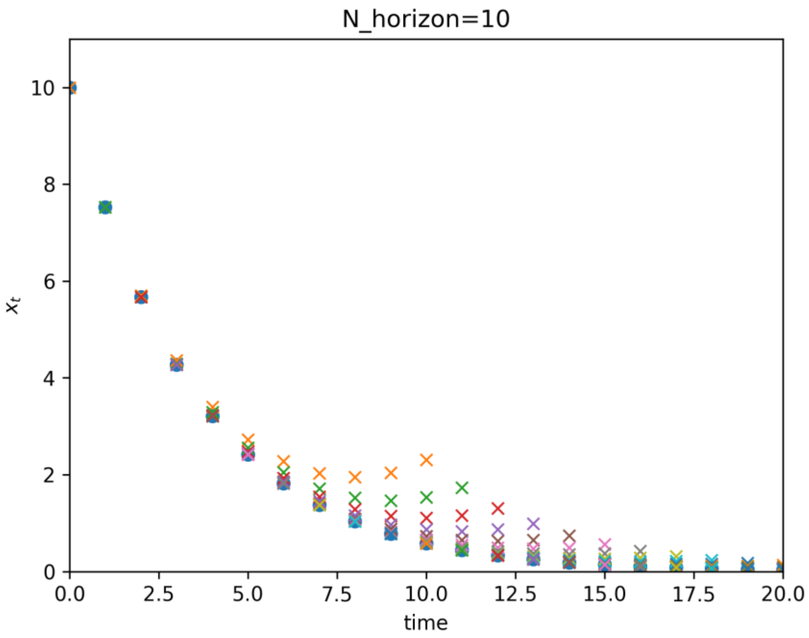
What is the usual practical solution?

- Choose *N* "large enough"
  - Can show: stability guaranteed for *N* large enough, but difficult/conservative to compute this limit
  - So what is "large enough" in practice? Rule of thumb: longer than "dominating dynamics"
  - (…but not too large, as large *N* might give robustness issues…)

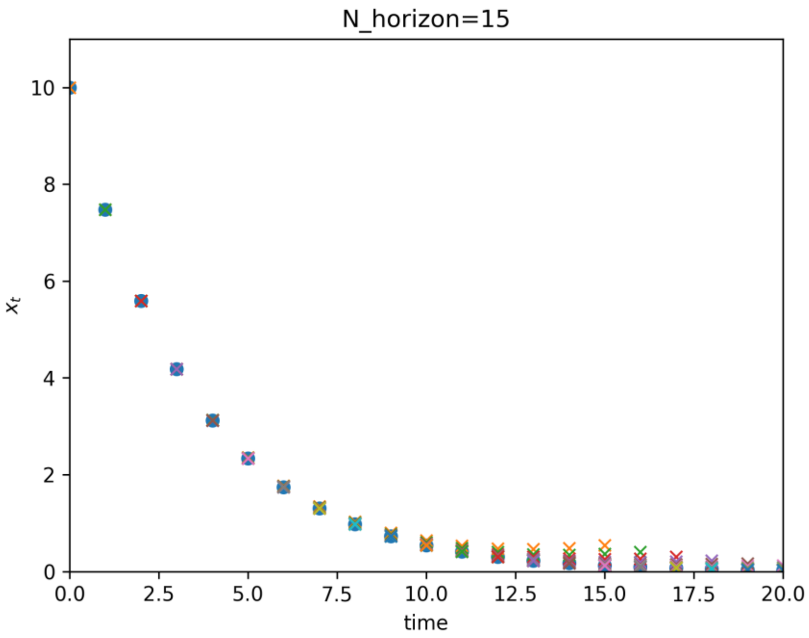
Open-Loop vs Closed-Loop: N = 5



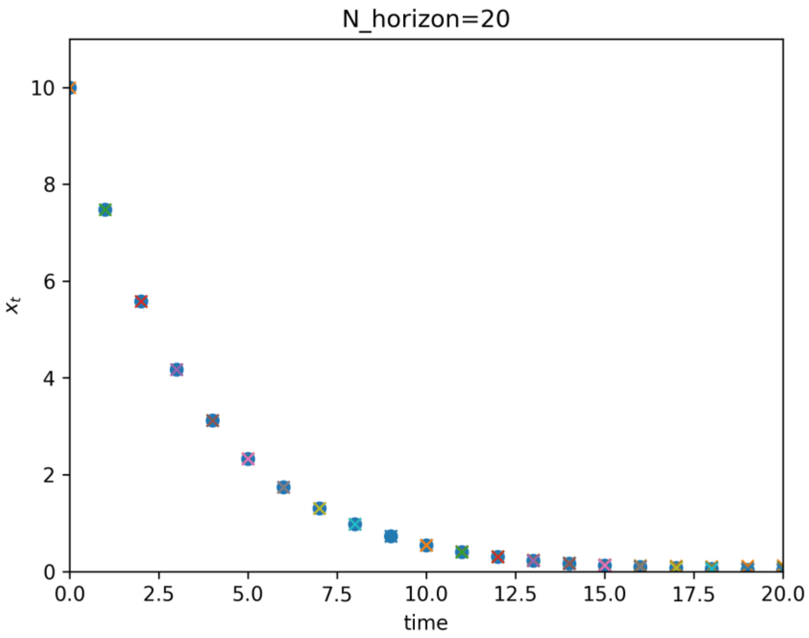
Open-Loop vs Closed-Loop: N = 10



Open-Loop vs Closed-Loop: N = 15



Open-Loop vs Closed-Loop: N = 20



Note: This assumes no model-plant mismatch!

NTNU | Norwegian University of Science and Technology

Why MPC over PID control?

Advantages of MPC:

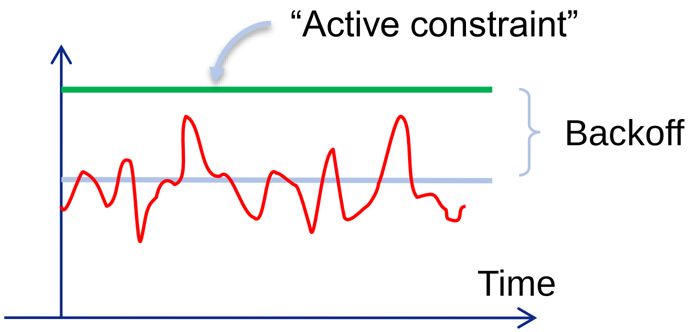
- **MPC handles constraints in a transparent way**
  - Physical constraints (actuator limits), performance constraints, safety limits, …
- Intuitive and easy to tune (…relatively, at least)
- MPC is by design multivariable (MIMO)
- MPC gives “optimal” performance (but what is the optimal objective?)

Disadvantage with MPC

- Online complexity (but only solving a QP, so not so bad)
- Requires models! Increased commisioning cost?
- Difficult to maintain?

“Squeeze and shift”

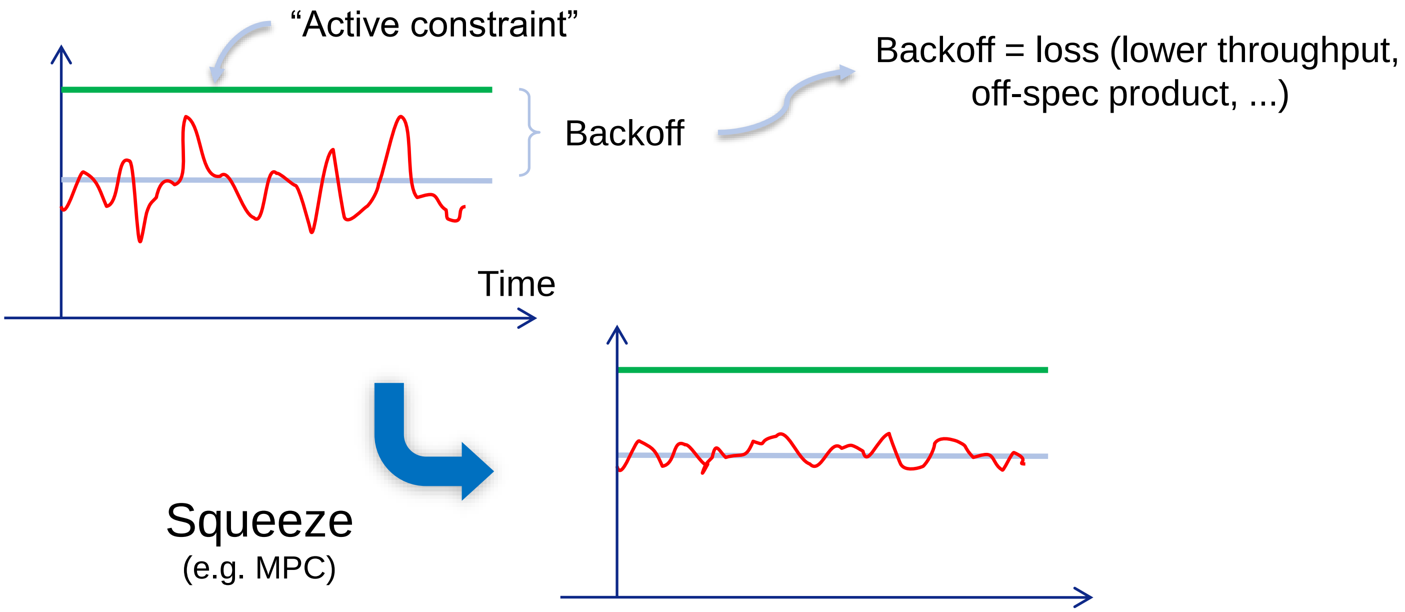
How MPC (or improved/advanced control in general) improves profitability



Backoff = loss (lower throughput, off-spec product, ...)

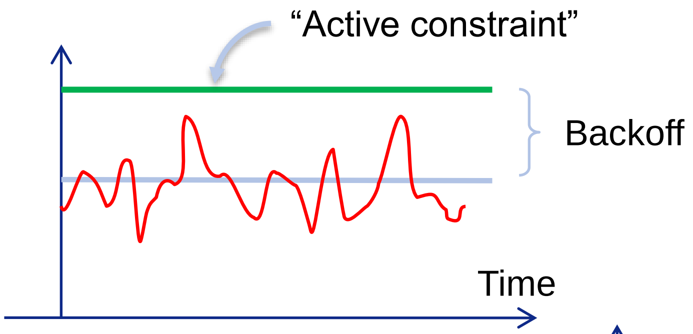
“Squeeze and shift”

How MPC (or improved/advanced control in general) improves profitability

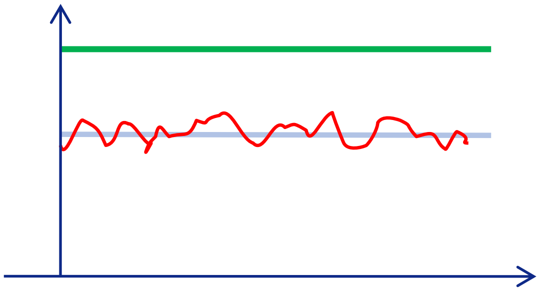


“Squeeze and shift”

How MPC (or improved/advanced control in general) improves profitability



Backoff = loss (lower throughput, off-spec product, ...)



Squeeze
(e.g. MPC)

Shift
(reduce backoff)