Computational Control

Model Predictive Control

Saverio Bolognani

Automatic Control Laboratory (IfA)
ETH Zurich

Optimal control beyond LQR

Beyond LQR

Constrained linear-quadratic finite-horizon optimal control

$$
\begin{aligned}
\min_{\substack{u_0,\dots,u_{T-1}\\ x_0,\dots,x_T}} \quad & \sum_{t=0}^{T-1} \left(x_t^\top Q x_t + u_t^\top R u_t \right) + x_T^\top S x_T \\
\text{subject to} \quad & x_{t+1} = A x_t + B u_t \quad \text{for } t = 0, \dots, T-1 \\
& x_0 = X_0 \\
& x_t \in \mathcal{X}_t \quad \text{(polytope)} \\
& u_t \in \mathcal{U}_t \quad \text{(polytope)}
\end{aligned}
$$

No backward induction, one single program with

- quadratic cost
- convex constraints
- linear equality constraints

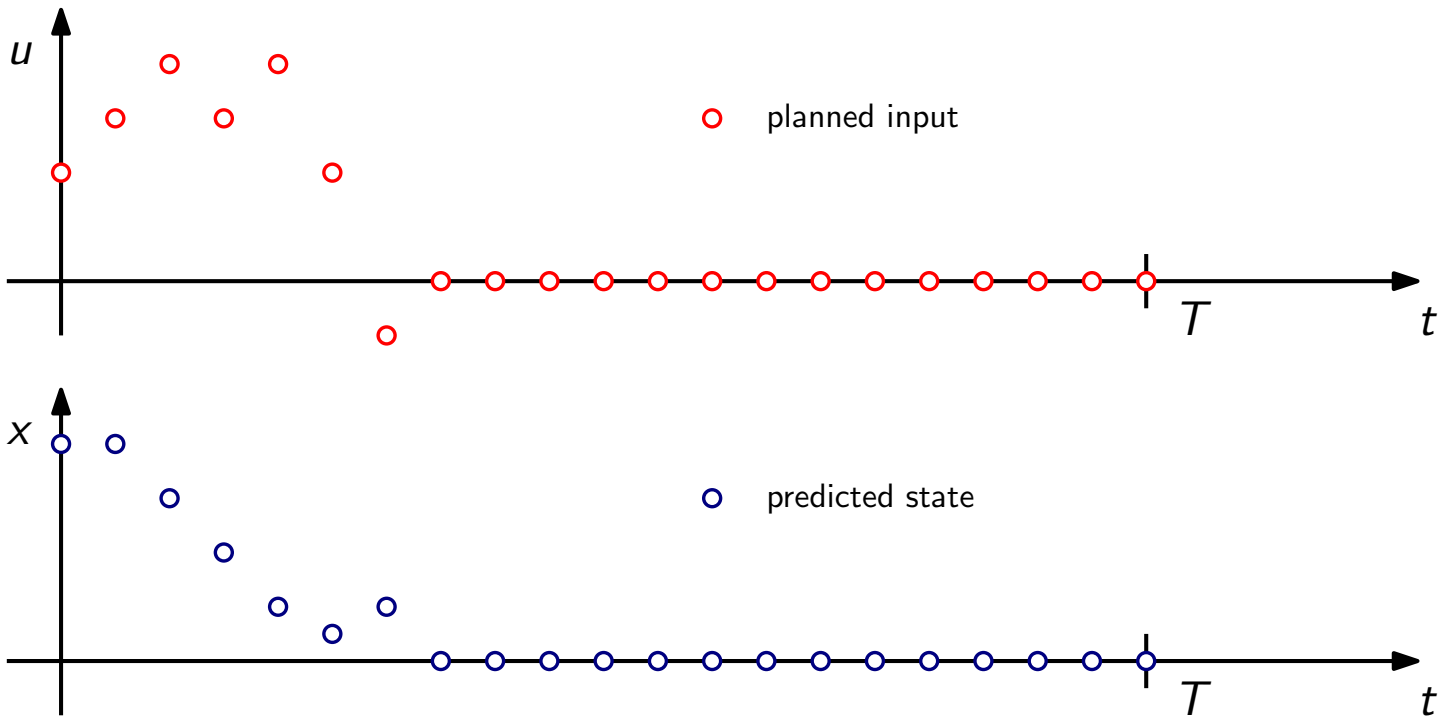
Open-loop optimal control

By solving the optimization **in one shot**, we get

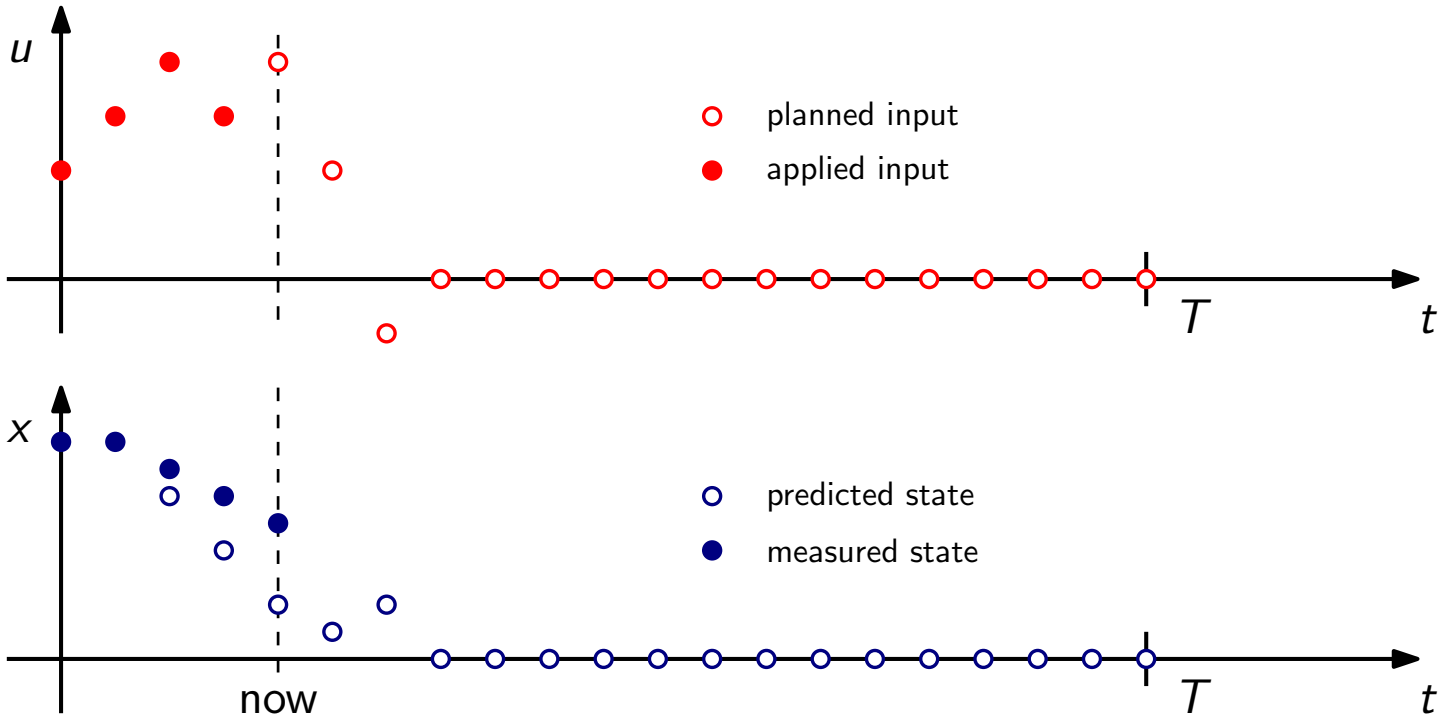
$$u_0^*, u_1^*, \ldots, u_{T-1}^* \in \mathbb{R}^m$$

instead of a state feedback

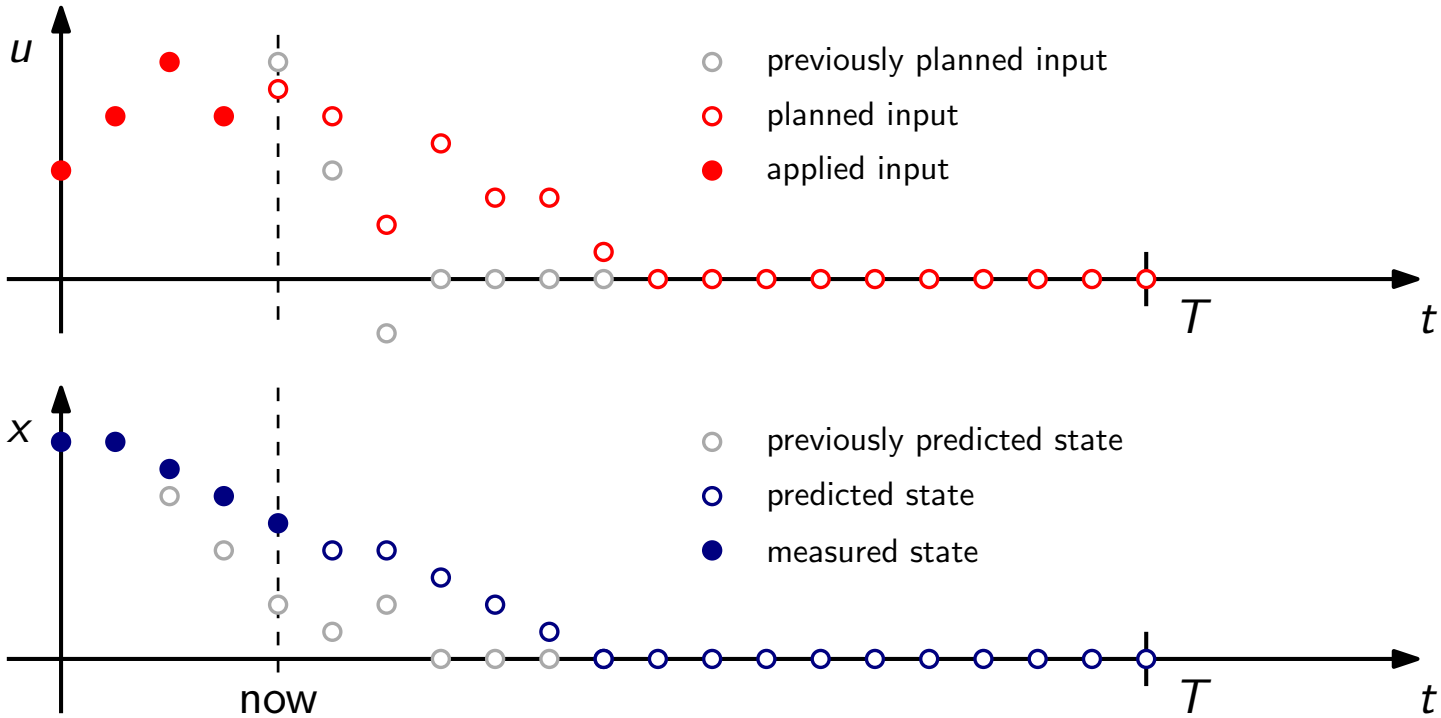
$$u_0(x), u_1(x), \ldots u_{T-1}(x) \qquad \text{like} \qquad u_t = \Gamma_t x_t \text{ (LQR)}$$



$\rightarrow$ **open-loop** optimal control strategy



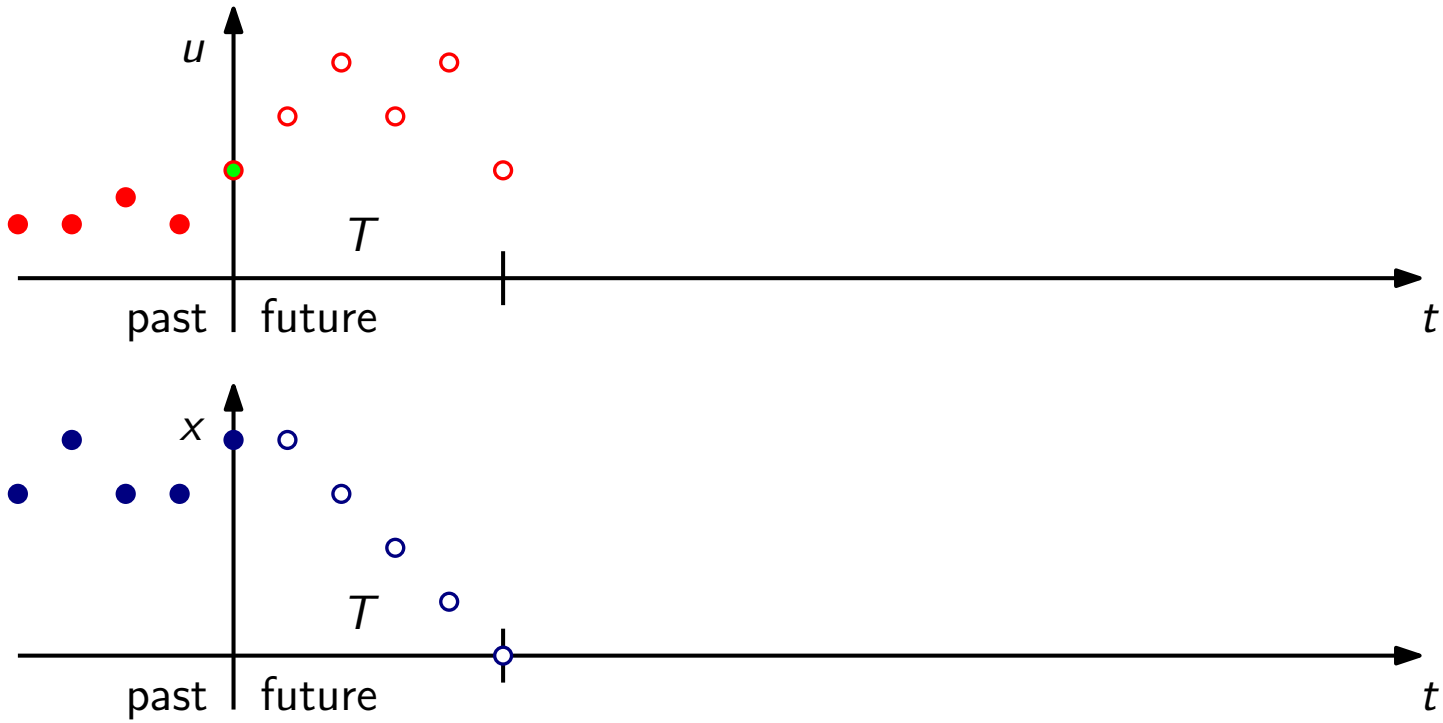
What if a **disturbance/model mismatch** takes you elsewhere?



Recompute!

- Incorporate new information about the current state
- Markovian system: the current state is all you need to know to compute the optimal control for the future
- Previously computed optimal trajectory is invalid
- Computation of the new optimal trajectory needs to happen in 1 time step

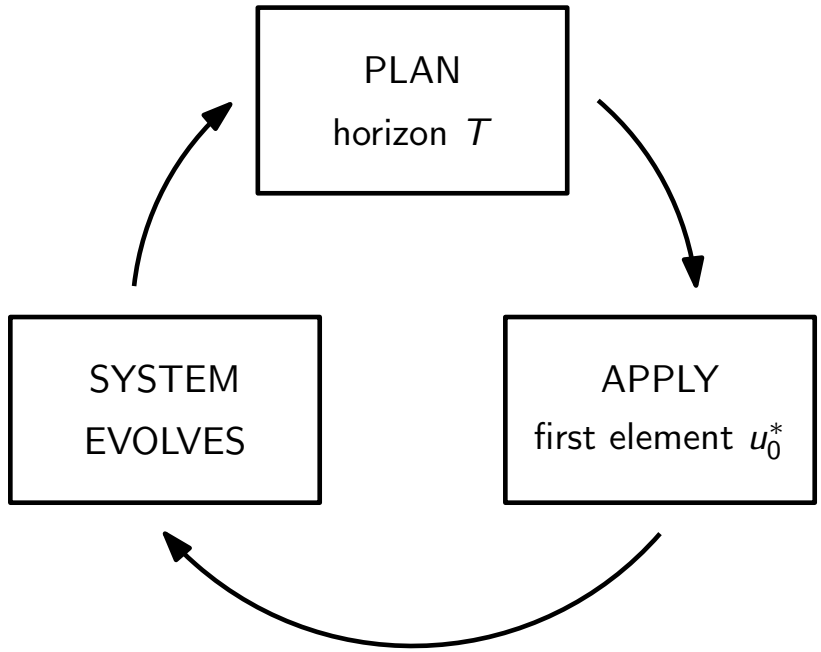
Receding horizon control



Iterative controller

- Apply only **one control input**, u_0^*
- Use new state measurement as new **initial conditions**
- Same horizon length $\rightarrow$ **receding horizon**
- If the system is time invariant $\rightarrow$ same **optimization problem**, different x_0

Receding horizon control



Iterative controller

- Apply only **one control input**, u_0^*
- Use new state measurement as new **initial conditions**
- Same horizon length $\rightarrow$ **receding horizon**
- If the system is time invariant $\rightarrow$ same **optimization problem**, different x_0

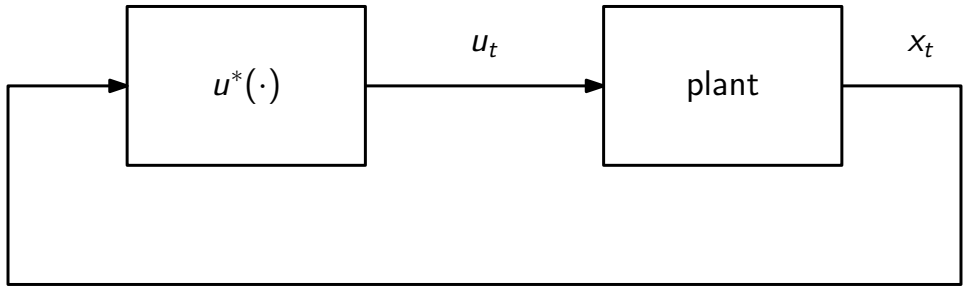
Receding-horizon control

The resulting **open-loop optimal control problem** corresponds to a *parametric optimization problem* parametrized in the initial state x.

Parametric optimization

$$
\begin{aligned}
u^*(x) \quad \text{determined by} \qquad \min_{\mathbf{u},\mathbf{x}} \quad & \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + x_T^\top S x_T \\
\text{subject to} \quad & x_{k+1} = A x_k + B u_k \\
& x_0 = x \\
& x_k \in \mathcal{X}_k \\
& u_k \in \mathcal{U}_k
\end{aligned}
$$

Feedforward or feedback?



What do we know about the optimization problem?

For the standard case of **quadratic cost** and **linear dynamics**

- The minimum cost is a continuous function of the initial condition x_0.
- Under **compactness** assumptions on the constraints, the parametric optimization problem has a solution when it's feasible.
- The map $x \mapsto u^*(x)$ is continuous

MPC control law

MPC is a **memory-less, nonlinear, time-invariant feedback control law**.

Question: Can I achieve zero tracking error with an MPC controller?

Does MPC solve the optimal control problem?

Example

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \sum_{t=0}^{T-1} a^t |u_t| \quad + |x(T)|^2 \qquad |a| < 1 \\ \text{subject to} \quad & x_{t+1} = x_t + u_t \end{aligned}$$

1. What is the optimal solution $\mathbf{x}, \mathbf{u}$?
2. What is the MPC control law $x \mapsto u^*(x)$?

The optimal trajectory defined by

- state cost function
- input cost function
- horizon length
- constraints

is never realized (even with no disturbance or model mismatch)!

$$\min_{\mathbf{u},\mathbf{x}} \quad \sum_{t=0}^{T-1} \left(x_t^\top Q x_t + u_t^\top R u_t \right) + x_T^\top S x_T, \qquad Q, S \succeq 0,\ R \succ 0$$

$$\text{subject to} \quad x_{t+1} = A x_t + B u_t$$

Finite-time LQR

- Backward computation of $P_T, P_{T-1}, \dots P_1$
- Linear feedback $u_t = \Gamma_t x_t$ with $\Gamma_t = -(R + B^\top P_{t+1} B)^{-1} B^\top P_{t+1} A$

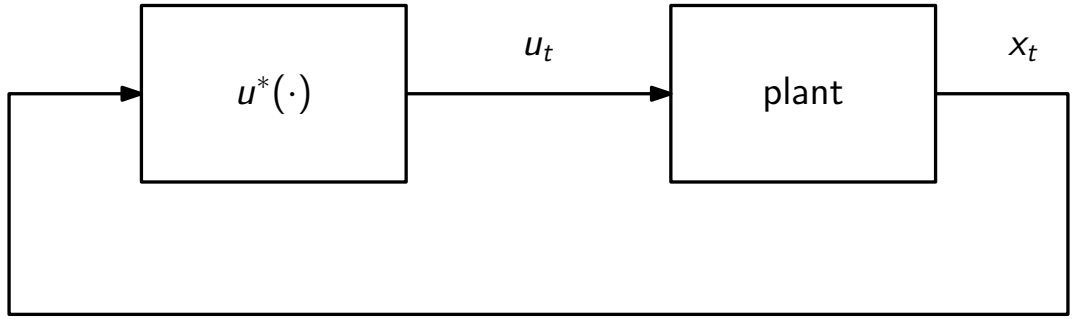
$$u_0^{\text{LQR}} = -(R + B^\top P_1 B)^{-1} B^\top P_1 A x_0$$
$$u_1^{\text{LQR}} = -(R + B^\top P_2 B)^{-1} B^\top P_2 A\, x_1$$

MPC

- Backward computation of $P_T, P_{T-1}, \dots P_1$
- Linear **time-invariant** feedback $u_t = \Gamma_0 x_t$

$$u_0^{\text{MPC}} = -(R + B^\top P_1 B)^{-1} B^\top P_1 A x_0 = u_0^{\text{LQR}}$$
$$u_1^{\text{MPC}} = -(R + B^\top P_1 B)^{-1} B^\top P_1 A\, x_1$$

Implementation of MPC



Option 1: Compute the **function** $u^*(\cdot)$ offline

- extremely complex with the exception of very few cases
- abundant offline computation power
- large memory requirement

Option 2: Compute the **value** of the function $u^*(\cdot)$ online at every iteration

- often a tractable optimization problem
- limited time to perform the computation online

Option 1: Explicit computation of u^*

We already know one case.

Unconstrained linear MPC

$\rightarrow$ first element of the finite-time LQR solution $\rightarrow$ **linear feedback** $u^*(\cdot)$

How about constraints?

Linear MPC + linear constraints*

$$
\begin{aligned}
\min_{\mathbf{u},\mathbf{x}} \quad & \sum_{k=0}^{T-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + x_T^\top S x_T, \qquad Q, S \succeq 0,\ R \succ 0 \\
\text{subject to} \quad & x_{k+1} = A x_k + B u_k \\
& x_0 = x \\
& \mathbf{x} \in \mathcal{P}_{\mathbf{x}} \quad \text{(polytope } C\mathbf{x} \leq d\text{)} \\
& \mathbf{u} \in \mathcal{P}_{\mathbf{u}} \quad \text{(polytope } E\mathbf{u} \leq f\text{)}
\end{aligned}
$$

$*$ Extremely common MPC problem formulation

Example

Consider the **toy problem** (time horizon 1, scalar integrator)

$$\begin{aligned} \min_{u_0, x_1} \quad & u_0^2 + x_1^2 \\ \text{subject to} \quad & x_1 = x_0 + u_0 \\ & x_1 \leq 1 \end{aligned}$$

- Quadratic problem with linear constraints
- Parametrized in x_0
- The equality constraint (model) allows to eliminate x_1
- Convex $\Rightarrow$ KKT conditions are necessary and sufficient*

KKT conditions (only 1 inequality constraint)

$$\begin{aligned} \min_x \ & f(x) \\ \text{subject to } & g(x) \leq 0 \end{aligned} \quad \rightarrow \quad \begin{cases} \nabla f(x) + \mu \nabla g(x) = 0, & \mu \geq 0 \\ g(x) \leq 0 \\ \mu g(x) = 0 \quad \forall i \end{cases}$$

$$
\begin{aligned}
f(u_0) &= u_0^2 + (x_0 + u_0)^2 \\
g(u_0) &= x_1 - 1 = x_0 + u_0 - 1 \\
\nabla f(u_0) &= 4u_0 + 2x_0 \\
\nabla g(u_0) &= 1
\end{aligned}
$$

Let's consider the complementary slackness condition $\mu g(u_0) = 0$.

Case 1: $\mu = 0$ (and $g(u_0) \leq 0$)
We then have $\nabla f(u_0) = 0$, which means $u_0 = -x_0/2$.
With this input $g(u_0)$ then becomes $x_0/2 - 1$, so that $g(u_0) \leq 0$ means $x_0 \leq 2$.

If $x_0 \leq 2$ then $u_0^* = -x_0/2$.

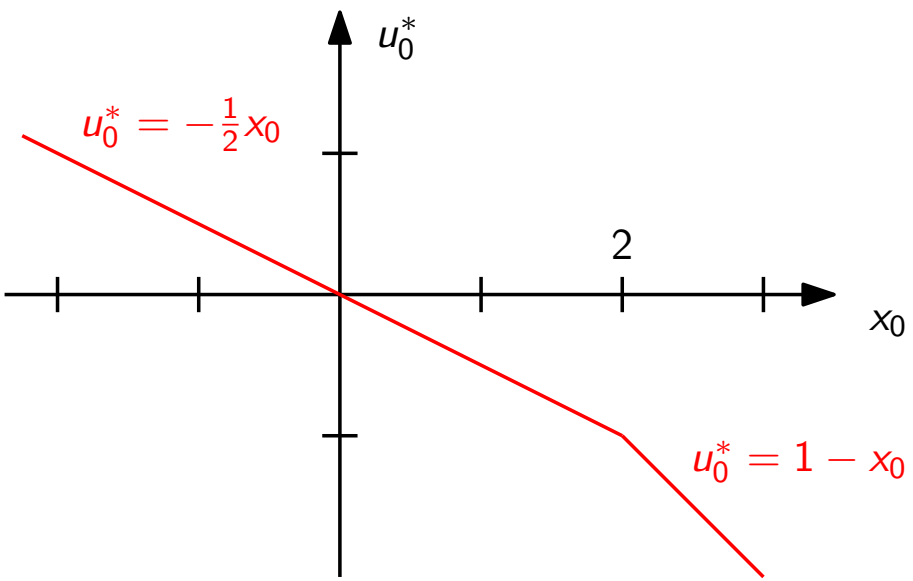
Case 2: $g(u_0) = 0$ (and $\mu \geq 0$)
$g(u_0) = 0$ implies that $u_0 = 1 - x_0$.
Because $\mu \geq 0$, we must have $\nabla f(u_0) \leq 0$ which means $4u_0 + 2x_0 \leq 0$.
With $u_0 = 1 - x_0$ this means $4 - 2x_0 \leq 0$, that is $x_0 \geq 2$.

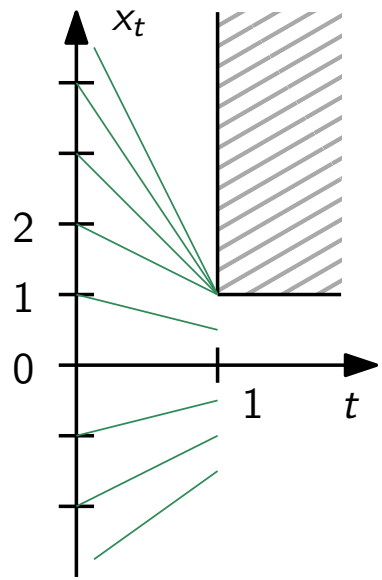
If $x_0 \geq 2$ then $u_0^* = 1 - x_0$.

Piecewise-affine feedback

$$\begin{aligned} \min_{u_0, x_1} \quad & u_0^2 + x_1^2 \\ \text{subject to} \quad & x_1 = x_0 + u_0 \\ & x_1 \leq 1 \end{aligned}$$

- $u^*(\cdot)$ is continuous
- $u^*(\cdot)$ is piecewise affine
- The number of affine regions depends on the number of constraints (how?)
- In each region, the same constraints are active/inactive

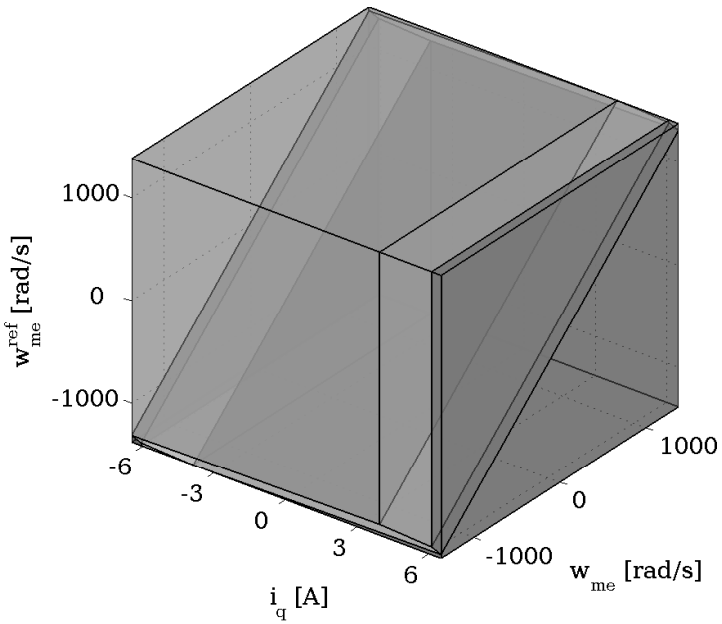


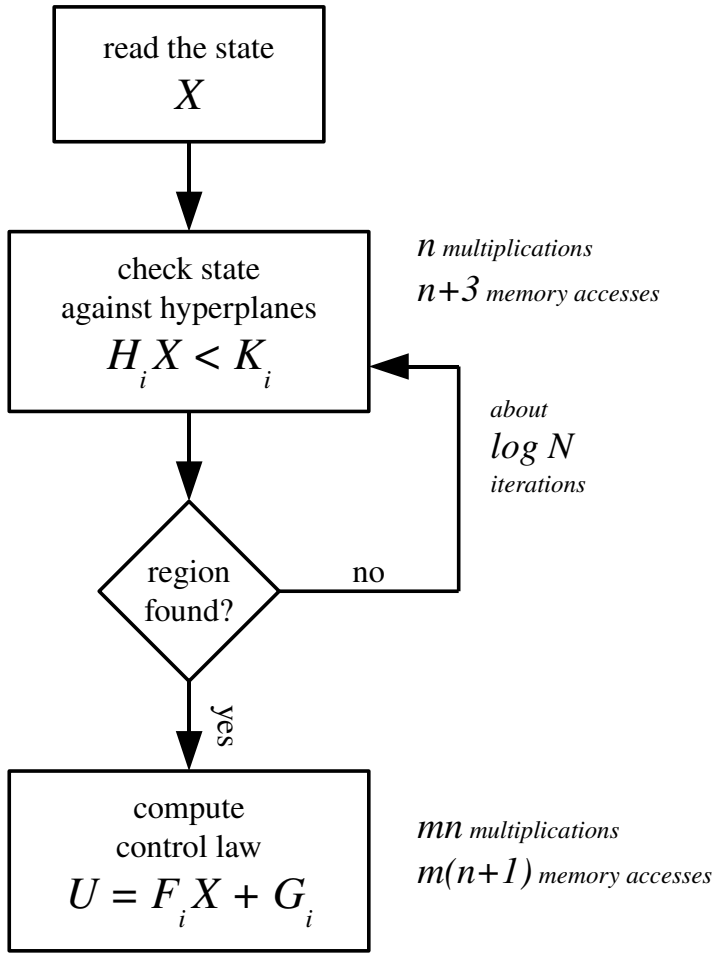


Explicit MPC in practice

- Tens or hundreds of constraints
- exponentially many regions
- $\rightarrow$ very long **offline optimization**
- even searching for the region is challenging in real-time!
- $\rightarrow$ even longer **offline preprocessing** (binary search tree)

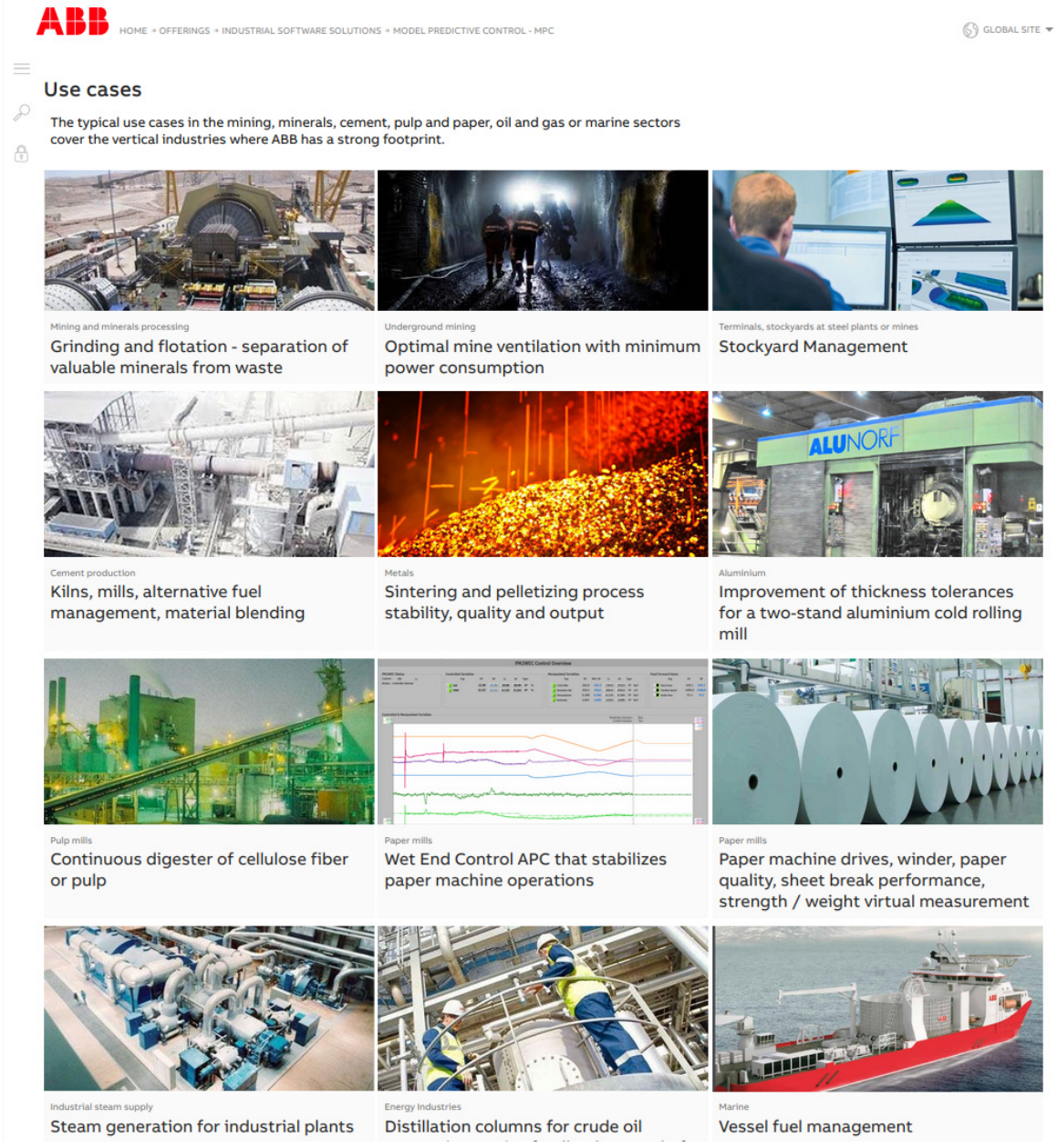
$\rightarrow$ 20 ms for $T = 5$, $n = 3$ (15 years ago)



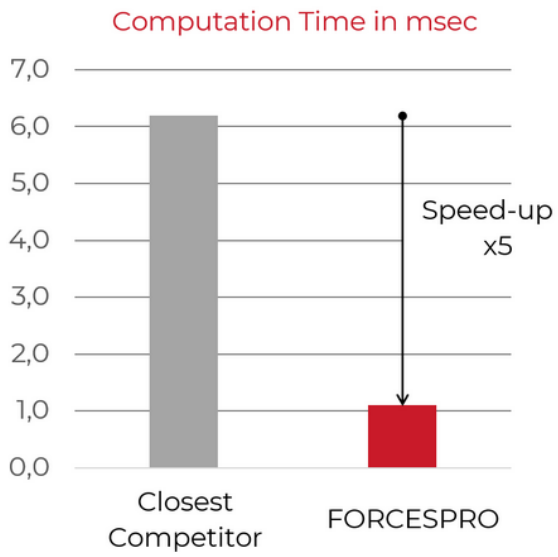


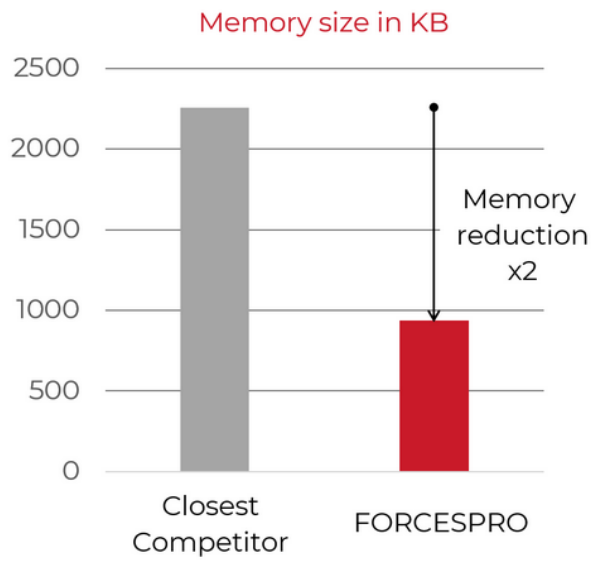
Multi-Parametric Toolbox (MPT) from ETH
`http://control.ee.ethz.ch/~mpt/`

Option 2: online computation



BENCHMARK OF A NONLINEAR MPC OPTIMIZATION PROBLEM FOR A DC MOTOR WITH PREDICTION HORIZON N=105





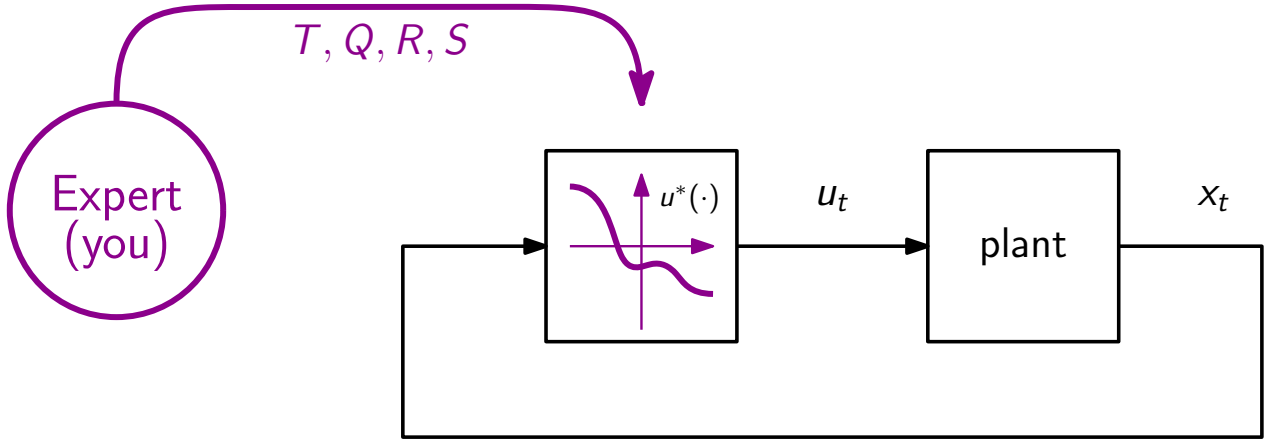
from `www.embotech.com`, ETH spin-off

Online MPC is increasingly common!

- custom optimization solvers
- approximate solutions of the optimization problem
- distributed solvers
- faster hardware

Stability

MPC as a feedback design tool



Ultimately, MPC is just a "clever" (principled) way to design static, time-invariant, non-linear feedback laws.

Closed-loop stability

- **Analysis:** compute u^* (hard!) and assess stability
- **Design:** derive conditions on T, Q, R, S that guarantee closed-loop stability

Tools:
Small-signal (local) stability?
Large-signal (global) stability?

Lyapunov analysis

Stable equilibrium

The equilibrium $x = 0$ is stable for the dynamics $x_{t+1} = f(x_t)$ if a small perturbation of the state perturbs the subsequent state trajectory in a continuous manner.

$$\forall \epsilon > 0, \exists \delta > 0 \text{ such that } \|x_0\| < \delta \Rightarrow \|x_t\| < \epsilon \; \forall t \geq 0.$$

Asymptotically stable equilibrium

The equilibrium $x = 0$ is asymptotically stable if it is stable and $\lim_{t \to \infty} \|x_t\| = 0$.



Lyapunov theorem

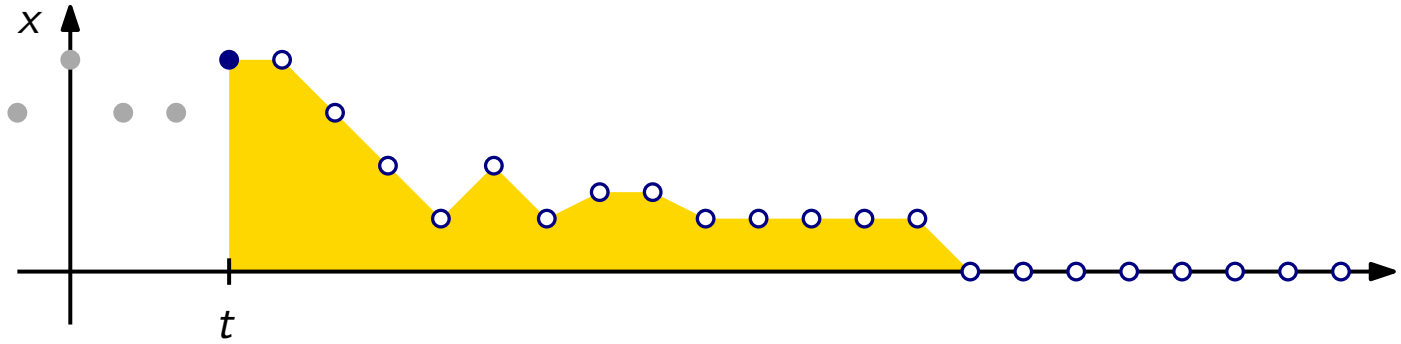
Let W be a real-valued function such that

$$W(0) = 0 \text{ and } W(x) > 0 \quad \forall x \neq 0$$
$$W(f(x)) < W(x) \quad \forall x \neq 0$$

Then $x = 0$ is asymptotically stable.

→ https://arxiv.org/abs/1809.05289

Stability of infinite horizon MPC

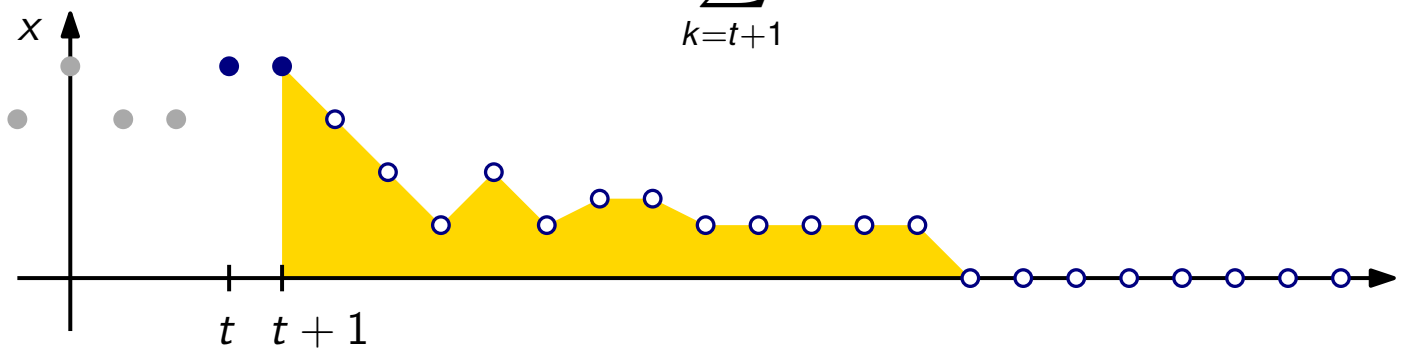


At time t, the MPC controller determines a trajectory $\mathbf{u}^*$ that minimizes

$$V_t^\infty(\mathbf{x}, \mathbf{u}) = \sum_{k=t}^{\infty} g_k(x_k, u_k)$$

By Bellman's principle, this trajectory minimizes also the "tail" cost

$$V_{t+1}^\infty(\mathbf{x}, \mathbf{u}) = \sum_{k=t+1}^{\infty} g_k(x_k, u_k)$$



This makes $W(x) = \min_{\mathbf{u},\mathbf{x}} V_k^\infty$ an obvious candidate **Lyapunov function**, under very reasonable assumptions (for example on the functions g_s).

- it is zero only at the target equilibrium $x = 0$
- it decreases along the closed-loop trajectories of the system

$$W(x_{t+1}) = W(x_t) - g_t(x_t, u^*(x_t)) < W(x_t)$$

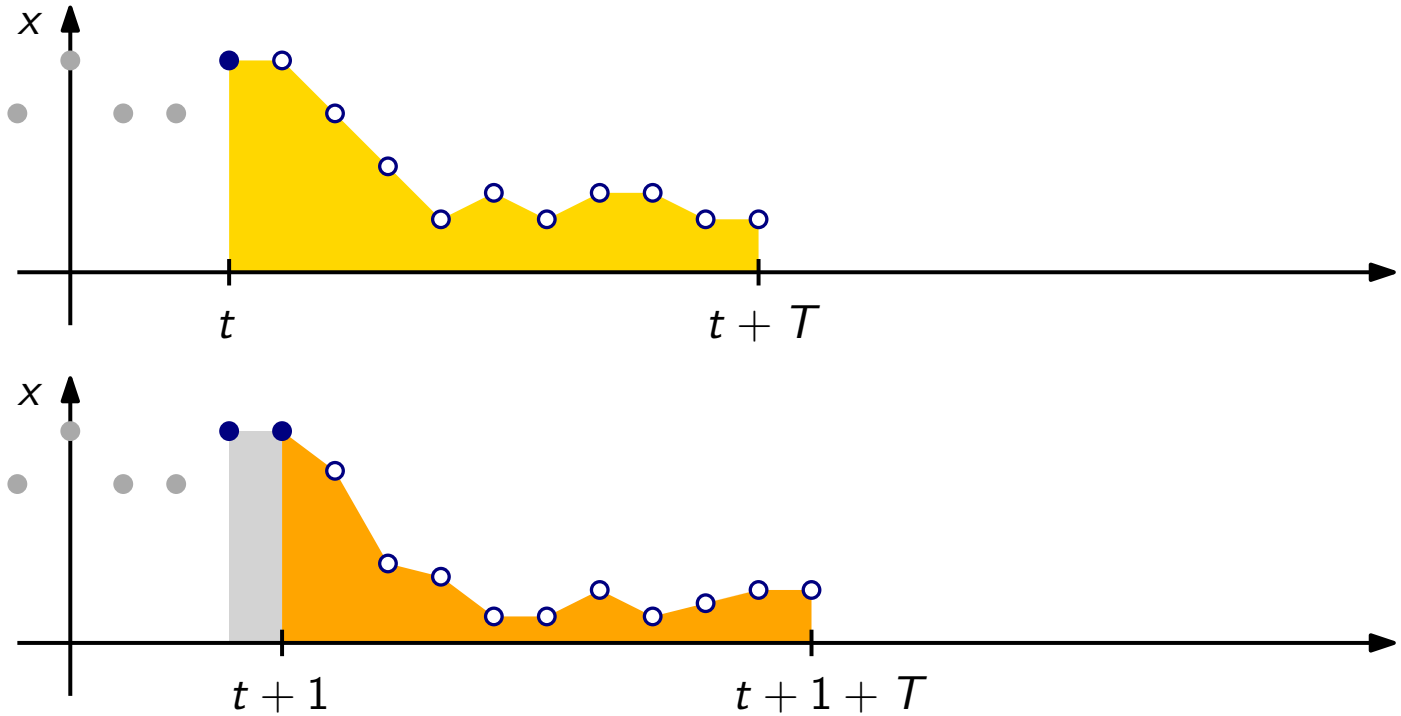
$\Rightarrow$ the equilibrium $x = 0$ is **asymptotically stable** for the closed-loop system.

Some subtleties

- Technically, we haven't defined the MPC control with an infinite-dimensional optimization problem
- We are assuming that the global minimum is found and attainable
- The functions g need to satisfy some properties
  (remember the observability assumptions on infinite-time LQR?)

Finite-time MPC stability

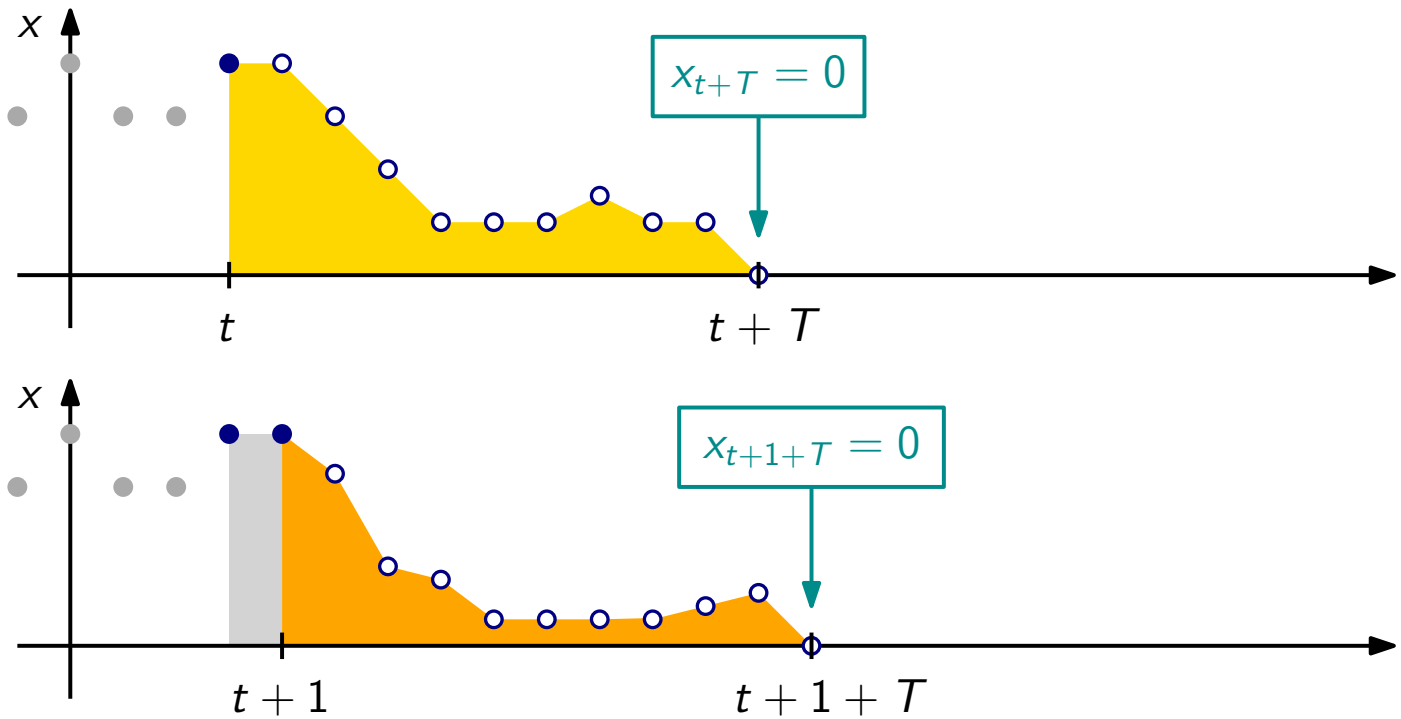
The same trick clearly does not apply to finite-time receding-horizon MPC.



- At $t + 1$, a different cost is minimized $\rightarrow$ different minimizer $\mathbf{x}, \mathbf{u}$
- The first element of the cost is removed, but a new element enters the tail.

Stability with terminal constraint

$$
\begin{aligned}
W(x) = \quad & \min_{\mathbf{u},\mathbf{x}} \quad \sum_{k=0}^{K-1} g_k(x_k, u_k) \\
\text{subject to} \quad & x_{k+1} = f(x_k, u_k) \\
& x_0 = x, x_K = 0
\end{aligned}
$$



Theorem

The equilibrium $x = 0$ is asymptotically stable for the closed-loop dynamics generated by a finite-time MPC controller with **zero terminal constraint**.

$$W(x_{t+1}) = \min \sum_{k=t+1}^{t+T} g(x_k, u_k) = \min \left(\sum_{k=t}^{t+T-1} g(x_k, u_k) - g(x_t, u_t) + g(x_{t+T}, u_{t+T}) \right)$$

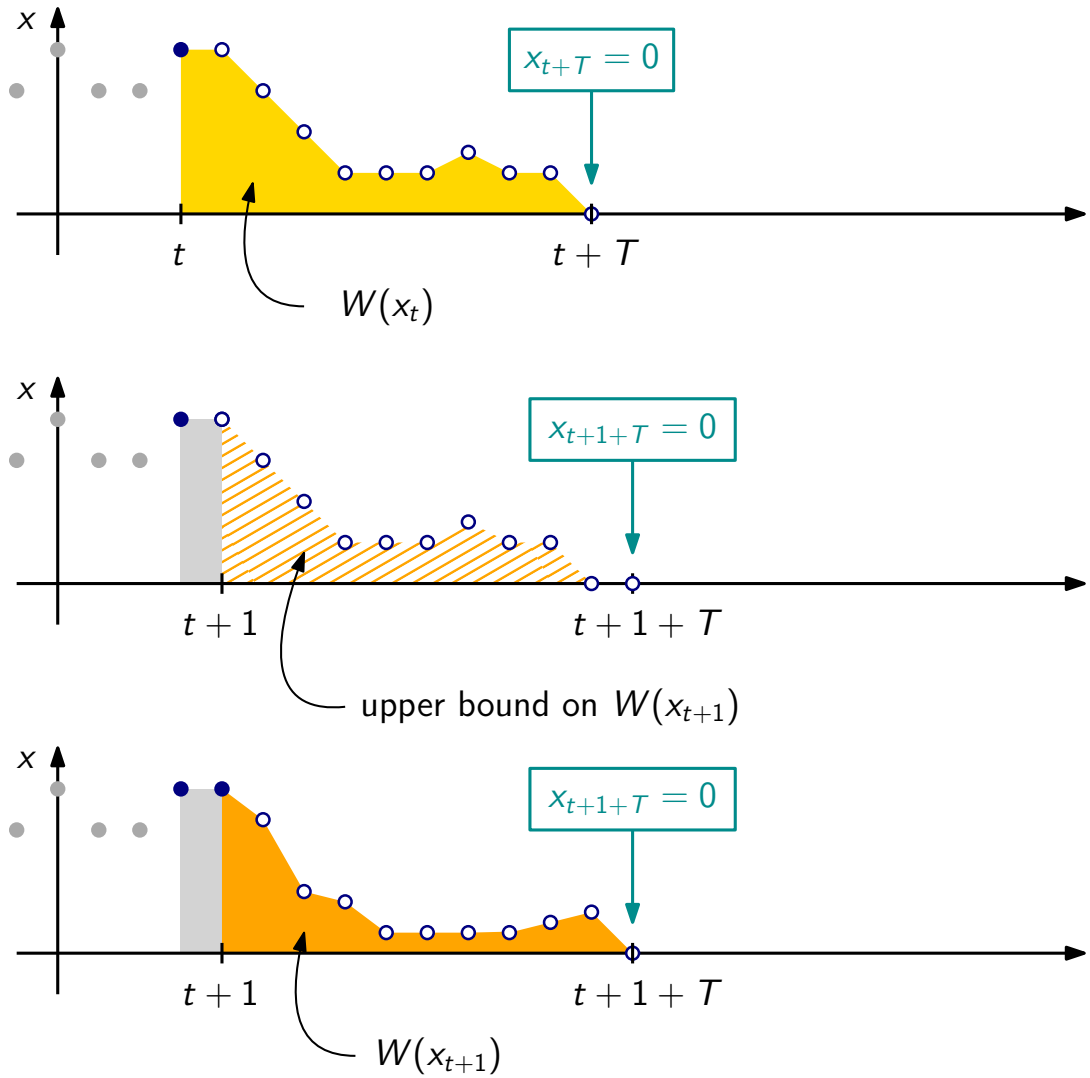
By definition of the minimum, this is small than the same quantity evaluated for an arbitrary u.
Let's evaluate it for the input that minimizes the optimization problem at time t, that is

$$\tilde{u} = \arg\min \sum_{k=t}^{t+T-1} g(x_k, u_k)$$

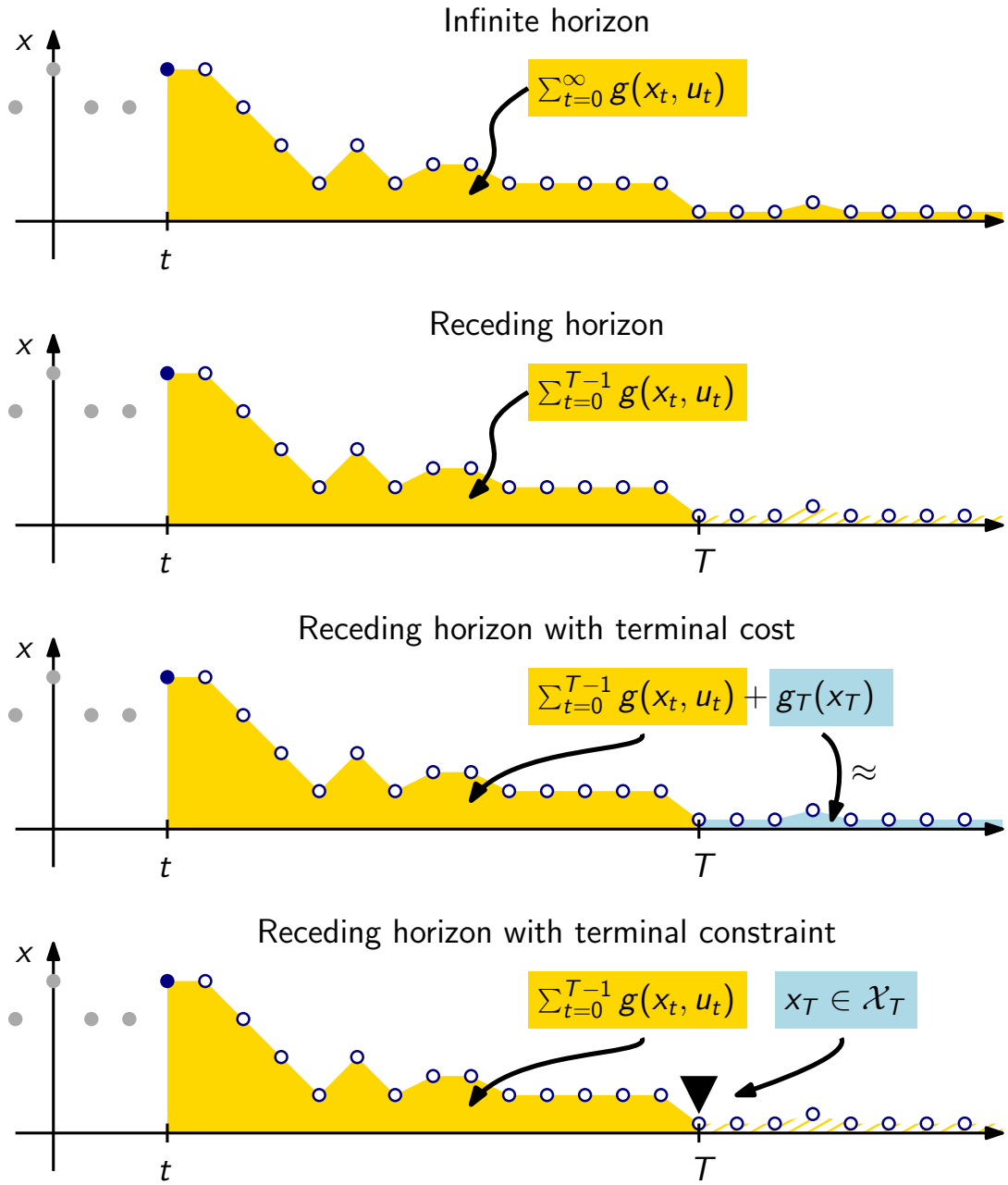
(which yields $x_{t+T} = 0$) followed by $u_{t+T} = 0$ that maintains the systems at $x_{t+T+1} = 0$. We then have

$$W(x_{t+1}) \leq W(x_t) - g(\tilde{x}_t, \tilde{u}_t) \leq W(x_t).$$

Key proof idea



Receding horizon as an approximation of infinite horizon



MPC closed loop stability

Many alternative stability conditions have been derived

- T large enough
- zero terminal constraint
- terminal region for the state
- ...

Subtleties

- We always assumed feasibility of the MPC problem, which may not be true (for example with a zero terminal constraint)
- Future feasibility of the problem may depends on the current control actions.
- We always assume that the global minimum can be found.

MPC minimizes a given cost function of the system state and input trajectories



MPC is just a clever way to design a static time-invariant feedback control law



MPC is a brute force numerical construction of Lyapunov functions



Disturbance rejection

Disturbance rejection specifications

Depending on the specifications of the control problem, we may want different level of **robustness** against disturbances.



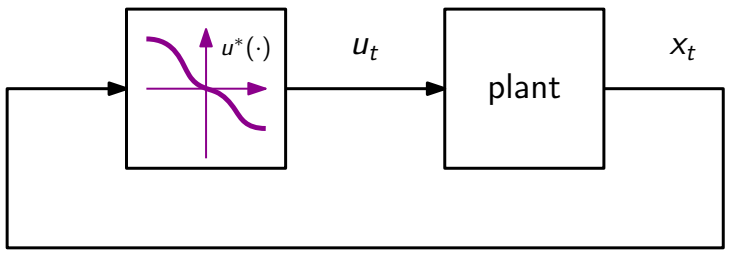


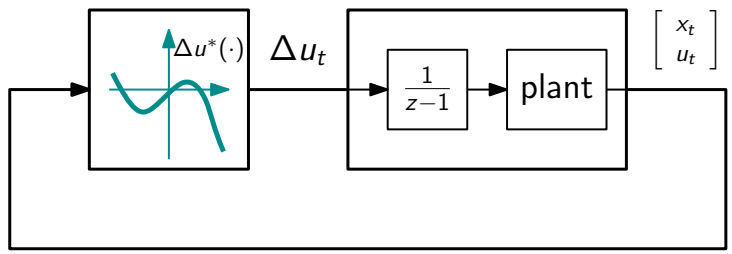


1. rejection at steady state: **zero offset**
2. rejection during the **transient**
3. guaranteed satisfaction of **constraint**

Incremental formulation of MPC (zero steady state offset)

Remember: MPC is "just" a well-designed non-linear static feedback law.



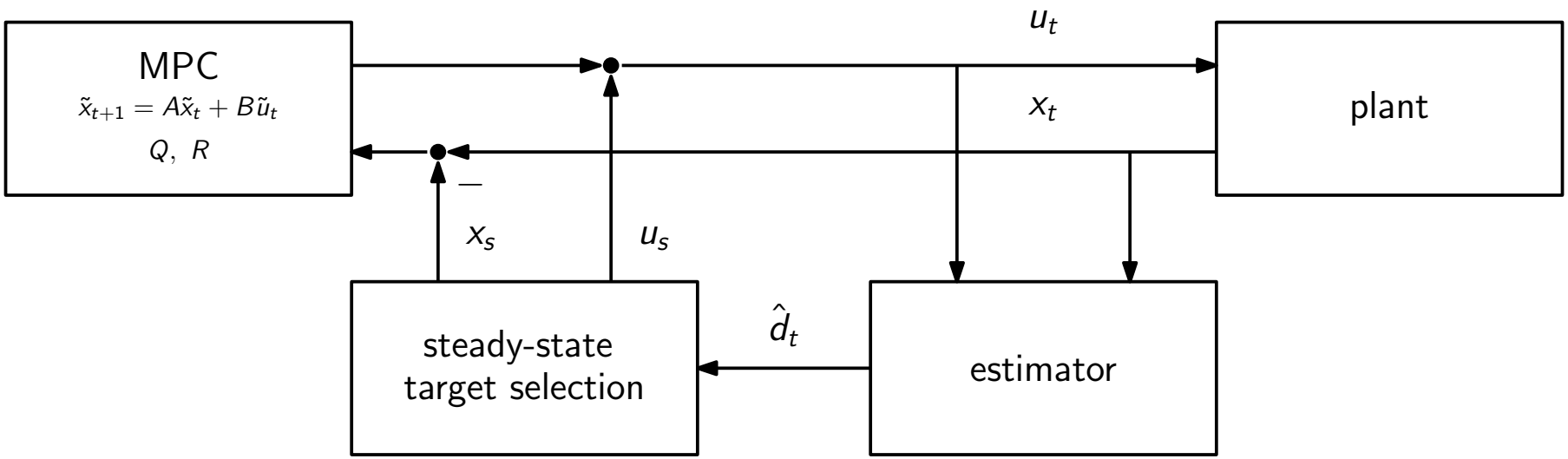


- If a non-zero input is necessary to drive the system to zero, then MPC will **not** achieve $x = 0$.
- Particularly fragile against
  - input disturbances
  - process noise
  - multiplicative noise uncertainty
- Very low input weights may cause numerical issues.

- $u_{t+1} = u_t + \Delta u_t$
- augment the plant with input integrator
  (if not present already!)
- integral nonlinear feedback law
- input weight penalizes changes in input signal

Disturbance rejection in MPC during transients

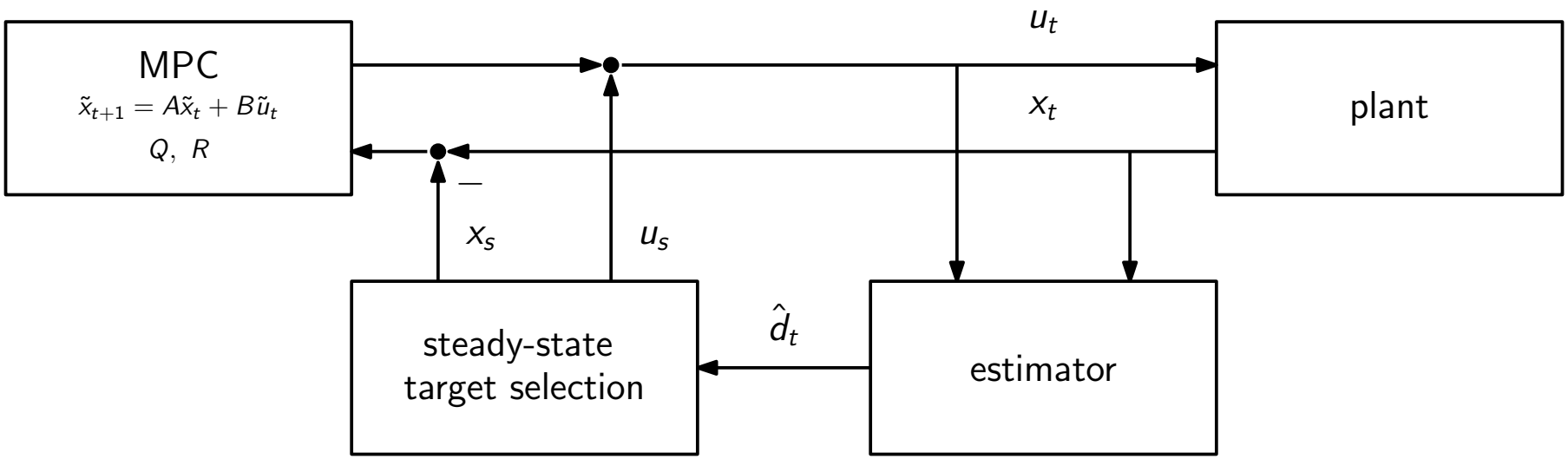
1. Based on transient measurements and plant model, **estimate the disturbance**
2. Correct the **steady-state target** based on the estimate of the disturbance



Unless better information is available, a constant model is often appropriate

$$d_{t+1} = d_t \qquad\qquad d_t \in \mathbb{R}^{n_d}, \quad n_d \leq n$$

Step 1: Estimate the disturbance



For example, a Luenberger observer

- **prediction** of x_{t+1} based on measurements x_t, u_t and estimate $\hat{d}_t$

$$\hat{x}_{t+1} = Ax_t + Bu_t + B_d\hat{d}_t$$

- **correction** based on the prediction error

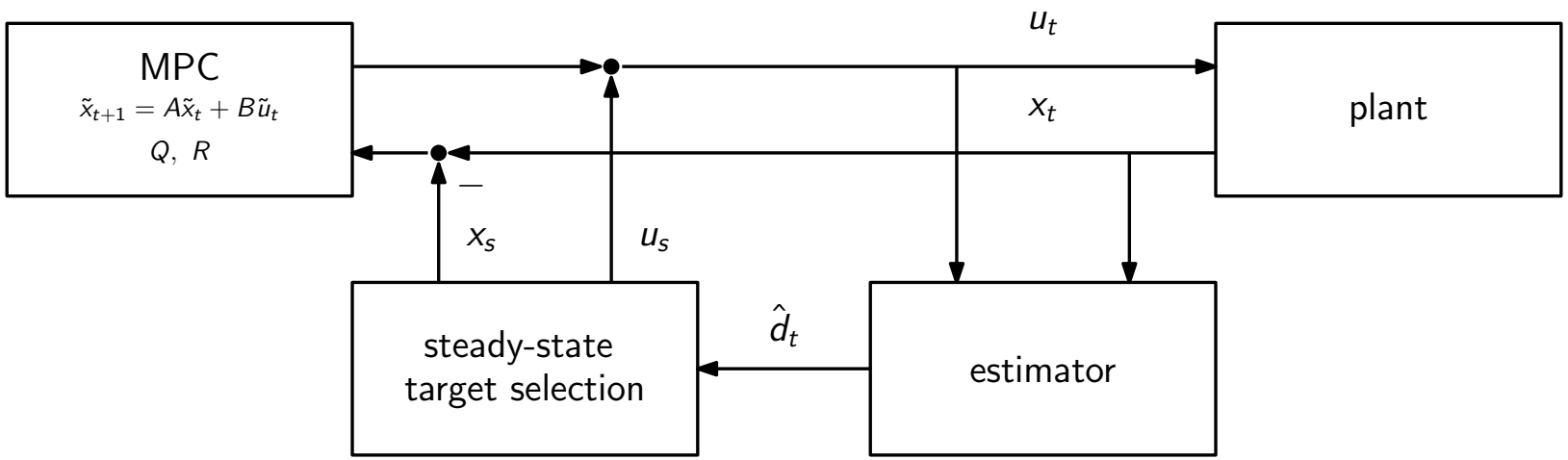
$$\hat{d}_{t+1} = \hat{d}_t + L(x_{t+1} - \hat{x}_{t+1}) \qquad \text{with } I - LB_d \text{ stable}$$

Step 2: Correct the steady-state target

Based on the constant disturbance model, $\hat{d}_k$ is our best estimate of the steady-state d_s.

Steady-state selection problem

$$\begin{aligned} \min_{x_s, u_s} \quad & \|x_s\|_{Q_s}^2 + \|u_s\|_{R_s}^2 \\ \text{subject to} \quad & \begin{bmatrix} I - A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = B_d \hat{d}_t \\ & x_s \in \mathcal{X} \\ & u_s \in \mathcal{U} \end{aligned}$$



Extra: advanced disturbance models

Good regulator theorem

"Every good regulator of a system must be a model of that system"

If a better model is available, use it!

- periodic disturbance? ramps?
- see Internal Model Principle
- the closed loop system will reject disturbances of the class that you modeled
- how many integrators do you need to reject a step disturbance? a ramp?

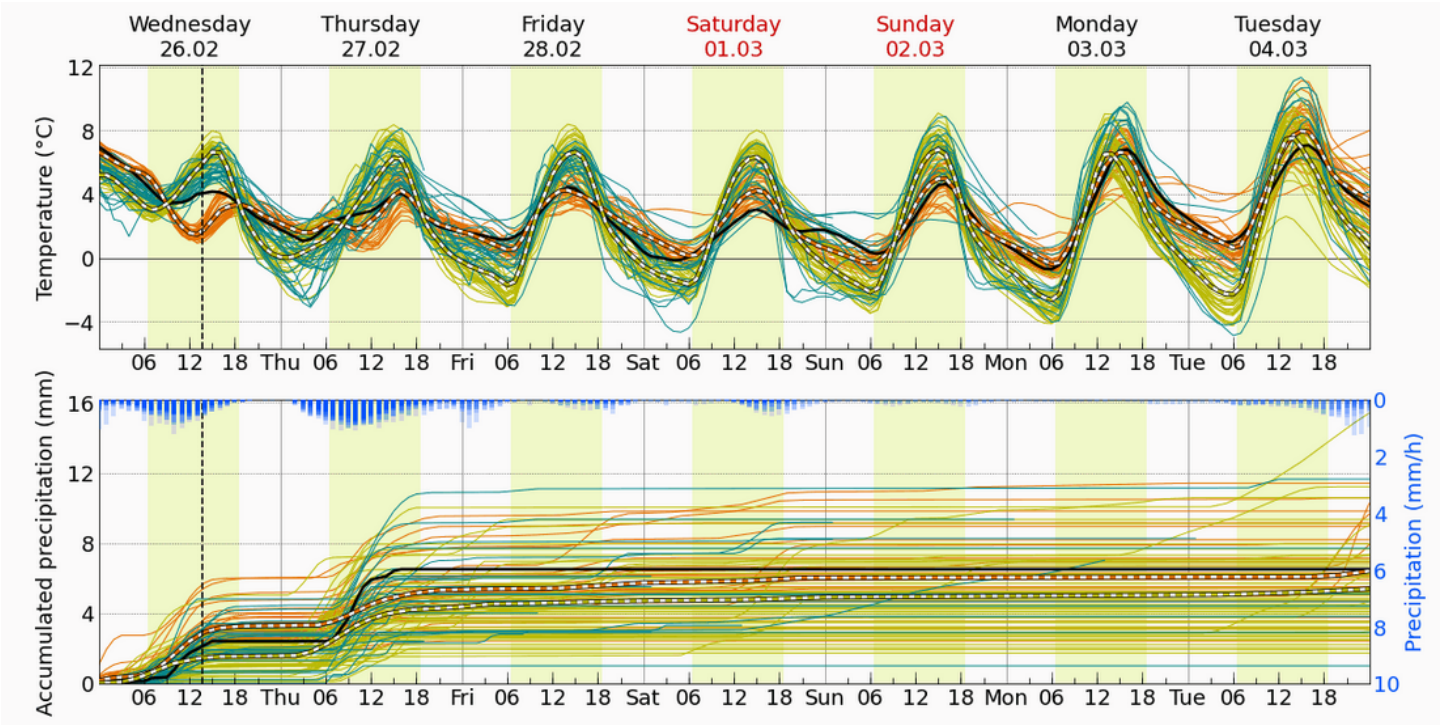
$$\begin{bmatrix} x_{t+1} \\ \hat{d}_{t+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x_t \\ \hat{d}_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t$$

If a model for the disturbance dynamics is available, it can be used

- in the disturbance estimator
- in the MPC model (instead of simply shifting the steady state targets)

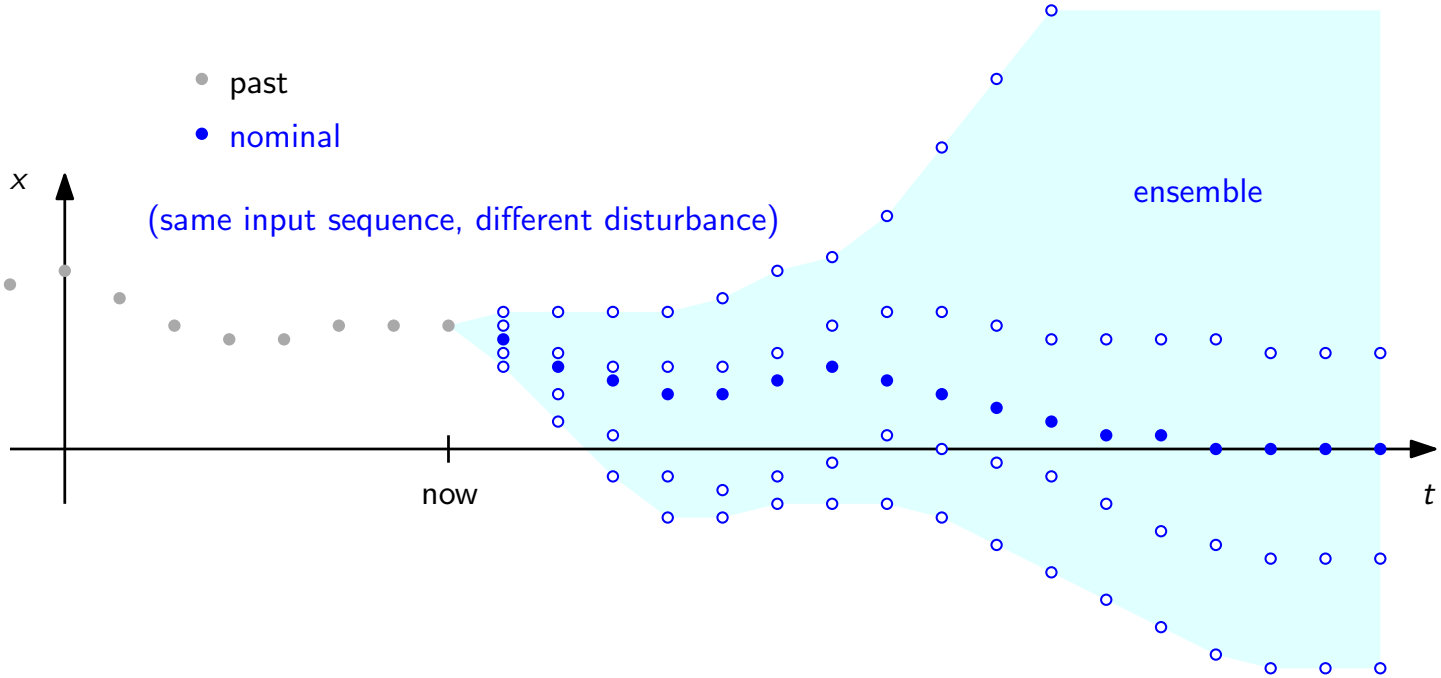
Robust satisfaction of constraints

- **Guaranteed satisfaction of constraints**
- Constraint satisfaction with high probability
- Constraint satisfaction for many samples (Monte Carlo, scenarios)
- Bound on expected violation (CVar - conditional value at risk)
- ...and more



Example: open field irrigation system with constraints on under-/over- watering.

We can do better than that if we have **prior information** on the disturbance.



$$x_{t+1} = f(x_t, u_t, w_t)$$

- **finite possible disturbances** $w_t \in \mathbb{W} = \{w^0, w^1, \ldots, w^p\}$
- disturbance set $w_t \in \text{co}\{w^0, w^1, \ldots, w^p\}$
- probability distribution $w_t \sim \mathcal{W}$

How can we use this prior information?

Finite possible disturbances

Linear system, additive disturbance

$$x_{t+1} = Ax_t + Bu_t + Dw_t$$

Robust satisfaction of constraints

Feedback control law $u^*(x)$ determined as the first step of the **open-loop robust optimal control problem**

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{x}} \quad & \sum_{k=0}^{T-1} g_k(x_k^{(\bar{w})}, u_k) \quad + g_T(x_K^{(\bar{w})}) \\
\text{subject to} \quad & \left\{ x_{k+1}^{(w)} = Ax_k^{(w)} + Bu_k + Dw_k \right\}_{w \in \mathbb{W}} \\
& x_0^{(w)} = x \quad \forall \mathbf{w} \in \mathbb{W} \\
& x_k^{(w)} \in \mathcal{X} \quad \forall k, \ \forall \mathbf{w} \in \mathbb{W} \\
& u_k \in \mathcal{U}
\end{aligned}$$

where $\bar{w} \in \mathbb{W}$ is a **nominal** disturbance, $\mathbb{W}$ is the ensemble of disturbances.

Example of unfeasible robust trajectory

A very simple system

$$x_{t+1} = x_t + u_t + w_t, \qquad |w_t| \leq 0.5$$

- $g(x,u) = x^2 + u^2$
- state constraint $|x_t| \leq 1$, unconstrained input
- horizon $T = 5$

Simple integration of the plant dynamics yields

$$|x_5| \leq 1 \qquad \Leftrightarrow \qquad -1 \leq x_0 + \sum_{k=1}^{4} u_k + \sum_{k=1}^{4} w_k \leq 1$$

If this needs to hold for all feasible disturbance w, we have

$$-x_0 - 1 - 2 \leq \sum_{k=1}^{4} u_k \leq -x_0 + 1 - 2 \qquad \text{for } w_k = 0.5, 0.5, 0.5, \ldots$$

$$-x_0 - 1 + 2 \leq \sum_{k=1}^{4} u_k \leq -x_0 + 1 + 2 \qquad \text{for } w_k = -0.5, -0.5, -0.5, \ldots$$

Example of unfeasible robust trajectory

A very simple system

$$x_{t+1} = x_t + u_t + w_t, \qquad |w_t| \leq 0.5$$

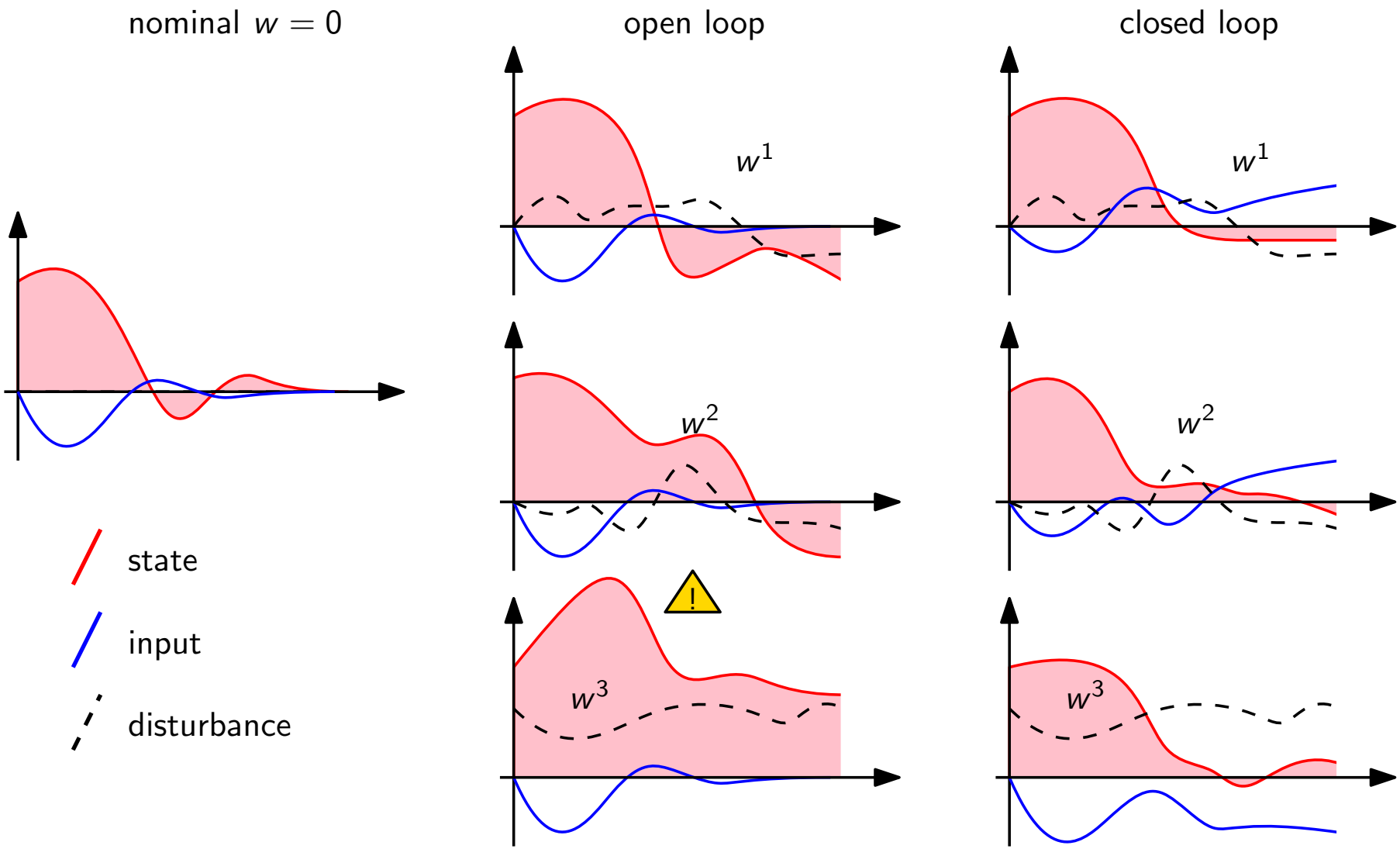
- cost $g(x, u) = x^2 + u^2$
- state constraint $|x_t| \leq 1$, unconstrained input
- horizon $T = 5$

A feasible (although suboptimal) proportional controller clearly exists:

$$u_t = -x_t$$

as it yields the closed loop dynamics $x_{t+1} = w_t$, which is clearly within bounds.

MPC was supposed to be a clever way to produce static time-invariant feedback laws, why is it failing to do so?


nominal $w = 0$
open loop
closed loop
w^1
w^2
w^3
!
state
input
disturbance

In solving the **open loop** finite-time optimal control at the core of the MPC routine, we are looking for a feasible **input sequence**

$$u_0, u_1, \ldots, u_K$$

that produces a feasible trajectory **regardless of the disturbance sequence** w_k.

That leads to

- extremely conservative control sequences
- infeasible problems

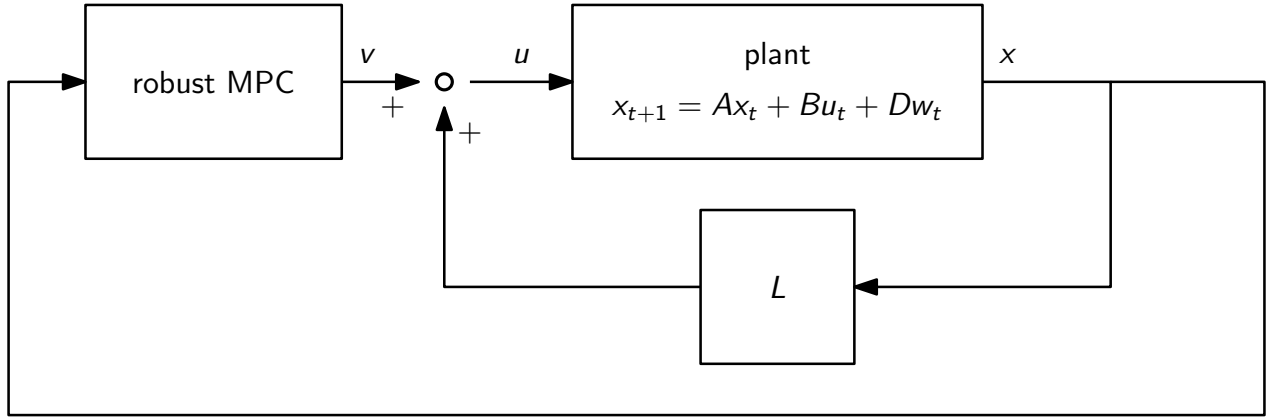
MPC vs closed loop optimal control

We would like to optimize over the set of **input policies**

$$u_k = \pi_k(x_k) \quad k = 0, \ldots, K-1$$

which is different from optimizing over open-loop policies, despite the receding horizon.

Nested controllers for robust MPC



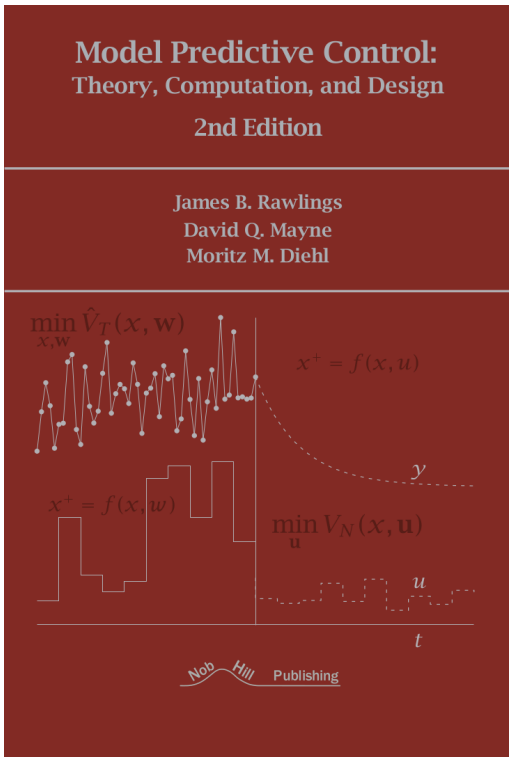
One way to decrease conservativeness of robust MPC is to "wrap" the plant around another controller (MPC or not).

$$x_{t+1} = Ax_t + B(v_t + Lx_t) + Dw_t = (A + BL)x_t + Bv_t + Dw_t$$

- The sequence v_k is an open-loop optimal policy (not a function of w_k) in the MPC optimization problem
- The feedback gain L rejects the disturbance w_k
  - $A + BL$ Hurwitz
  - design L **offline** for good disturbance rejection (with your preferred method)

Note: In case of no disturbance, there is no advantage in this parametrization, as v_k can include the term Lx_k computed for the nominal system trajectory.

And more...



- Output-feedback MPC
- Terminal constraints
- More powerful stability guarantees
- Examples

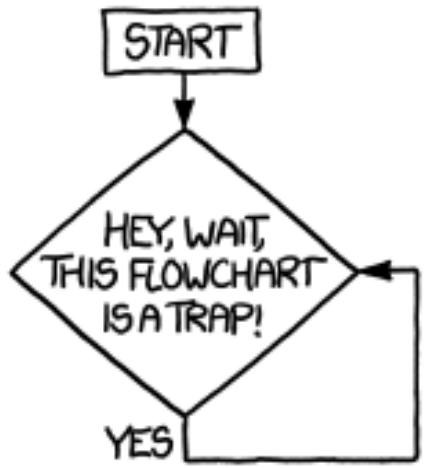
Chapters 1, 2, 7 of

Model Predictive Control:
Theory, Computation, and Design
James B. Rawlings, David Q. Mayne, Moritz M. Diehl
2nd edition, 2022

Free to download

`https://sites.engineering.ucsb.edu/~jbraw/mpc/`

The control engineer flowchart





This work is licensed under a
Creative Commons Attribution-NonCommercial 4.0 International License

https://bsaver.io/COCO