Linear Quadratic Regulation

Optimal control

$$u = K\mathbf{x}$$

$$J := \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dt \quad \Rightarrow \quad J = \int_0^\infty x(t)^T (Q + K^T R K) x(t)\, dt$$

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0. \qquad \dot{x} = (A + BK)x, \quad x(0) = x_0.$$

for a given K and x_0

$$x(t) = e^{(A+BK)t} x_0.$$

and

$$\begin{aligned} J &= \int_0^\infty x_0^T e^{(A+BK)^T t}(Q + K^T R K) e^{(A+BK)t} x_0\, dt \\ &= x_0^T \left(\int_0^\infty e^{(A+BK)^T t}(Q + K^T R K) e^{(A+BK)t}\, dt \right) x_0. \end{aligned}$$

Optimal control

$$J = x_0^T \left(\int_0^\infty e^{(A+BK)^T t} (Q + K^T RK) e^{(A+BK)t} \, dt \right) x_0.$$

J can be computed as

$$J = x_0^T X x_0$$

where X is the solution to the Lyapunov equation

$$(A+BK)^T X + X(A+BK) + Q + K^T RK = 0.$$

above equation can be rewriting in the form because

$$A^T X + XA - XBR^{-1}B^T X + Q + (XBR^{-1} + K^T)R(R^{-1}B^T X + K) = 0.$$

Optimal control

$$A^T X + XA - XBR^{-1}B^T X + Q + (XBR^{-1} + K^T)R(R^{-1}B^T X + K) = 0.$$

Note that K is confined to the term

$$(XBR^{-1} + K^T)R(R^{-1}B^T X + K) \succeq 0$$

$$\Longrightarrow \quad K = -R^{-1}B^T X.$$

and Algebraic Riccati Equation (ARE) in X

$$A^T X + XA - XBR^{-1}B^T X + Q = 0.$$

Infinite horizon LQR problem

discrete-time system $x_{t+1} = Ax_t + Bu_t$, $x_0 = x^{\text{init}}$

problem: choose $u_0, u_1, \ldots$ to minimize

$$J = \sum_{\tau=0}^{\infty} \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau \right)$$

with given constant state and input weight matrices

$$Q = Q^T \geq 0, \qquad R = R^T > 0$$

. . . an *infinite dimensional problem*

Infinite horizon LQR problem

problem: it's possible that $J = \infty$ for all input sequences $u_0, \ldots$

$$x_{t+1} = 2x_t + 0u_t, \qquad x^{\mathrm{init}} = 1$$

let's assume (A, B) is controllable

then for any x^{init} there's an input sequence

$$u_0, \ldots, u_{n-1},\ 0,\ 0, \ldots$$

that steers x to zero at $t = n$, and keeps it there

for this u, $J < \infty$

and therefore, $\min_u J < \infty$ for any x^{init}

Receding-horizon LQR control

consider cost function

$$J_t(u_t, \ldots, u_{t+T-1}) = \sum_{\tau=t}^{\tau=t+T} \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau \right)$$

- T is called *horizon*
- same as infinite horizon LQR cost, truncated after T steps into future

if $(u_t^*, \ldots, u_{t+T-1}^*)$ minimizes J_t, u_t^* is called (T-step ahead) *optimal receding horizon control*

in words:

- at time t, find input sequence that minimizes T-step-ahead LQR cost, starting at current time
- then use only the first input

Receding-horizon LQR control

example: 1-step ahead receding horizon control

find $u_t,\ u_{t+1}$ that minimize

$$J_t = x_t^T Q x_t + x_{t+1}^T Q x_{t+1} + u_t^T R u_t + u_{t+1}^T R u_{t+1}$$

first term doesn't matter; optimal choice for u_{t+1} is 0; optimal u_t minimizes

$$x_{t+1}^T Q x_{t+1} + u_t^T R u_t = (Ax_t + Bu_t)^T Q (Ax_t + Bu_t) + u_t^T R u_t$$

thus, 1-step ahead receding horizon optimal input is

$$u_t = -(R + B^T Q B)^{-1} B^T Q A x_t$$

. . . a constant state feedback

Receding-horizon LQR control

in general, optimal T-step ahead LQR control is

$$u_t = K_T x_t, \qquad K_T = -(R + B^T P_T B)^{-1} B^T P_T A$$

where

$$P_1 = Q, \qquad P_{i+1} = Q + A^T P_i A - A^T P_i B (R + B^T P_i B)^{-1} B^T P_i A$$

$i.e.$: same as the optimal finite horizon LQR control, $T - 1$ steps before the horizon N

- a constant state feedback
- state feedback gain converges to infinite horizon optimal as horizon becomes long (assuming controllability)

Closed-loop system

suppose K is LQR-optimal state feedback gain

$$x_{t+1} = Ax_t + Bu_t = (A + BK)x_t$$

is called *closed-loop system*

($x_{t+1} = Ax_t$ is called *open-loop system*)

is closed-loop system stable? consider

$$x_{t+1} = 2x_t + u_t, \qquad Q = 0, \qquad R = 1$$

optimal control is $u_t = 0x_t$, $i.e.$, closed-loop system is unstable

fact: if (Q, A) observable and (A, B) controllable, then closed-loop system is stable

Model Predictive Control

Introduction

모델예측 제어
(Model Predictive Control)

상태변수나 출력을 예측

↓

적절한 비용함수와 제약조건 설정

다양한 형태의 제약조건 만족

↓

최적화 실행

↓

제어입력 계산

Introduction

컴퓨터

이산시간 상태공간 모델

✓ Sampling Time 기준

연속적인 시간 → 불연속적 시간 → **계산 수행**

Analog **Digital**

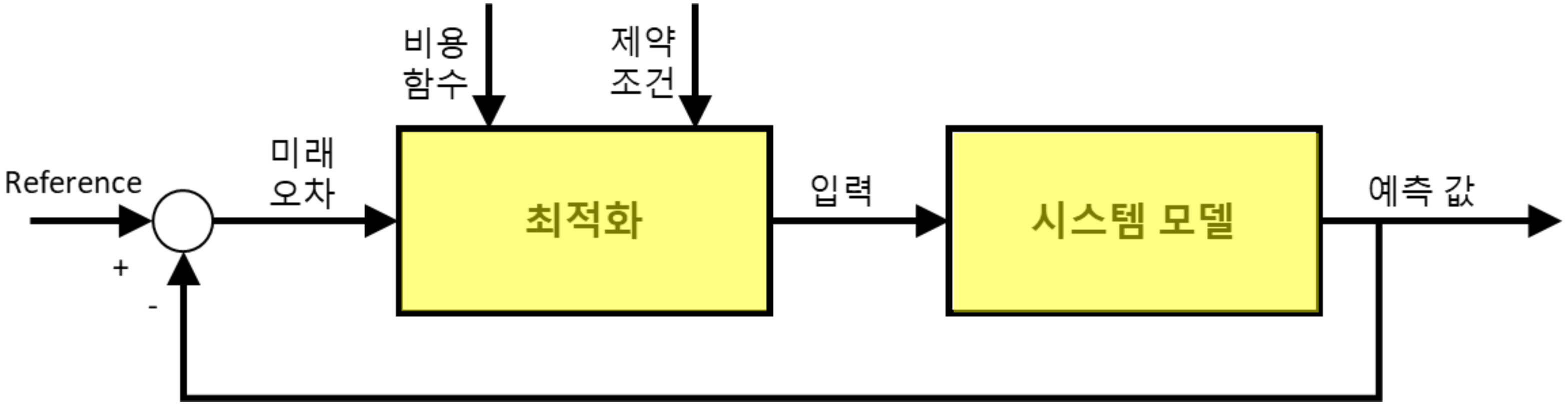
$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)$$

→ 이산시간

이산화 (Discretization)	연속적인 시간에서 나타내었던 시스템을 컴퓨터가 처리할 수 있도록 바꿔주는 과정
이산화된 시간 (Discrete-Time)	이산화된 시간
이산시간 상태공간 모델 (Discrete-Time State-Space Model	이산화된 시간에서 시스템에 상태공간 모델을 나타낸 것

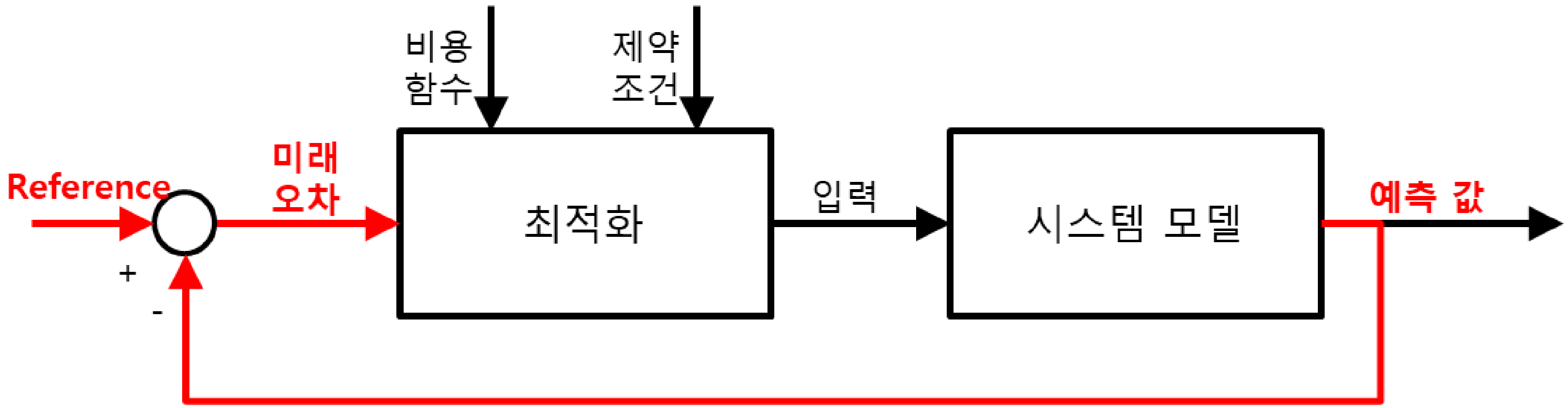
Introduction

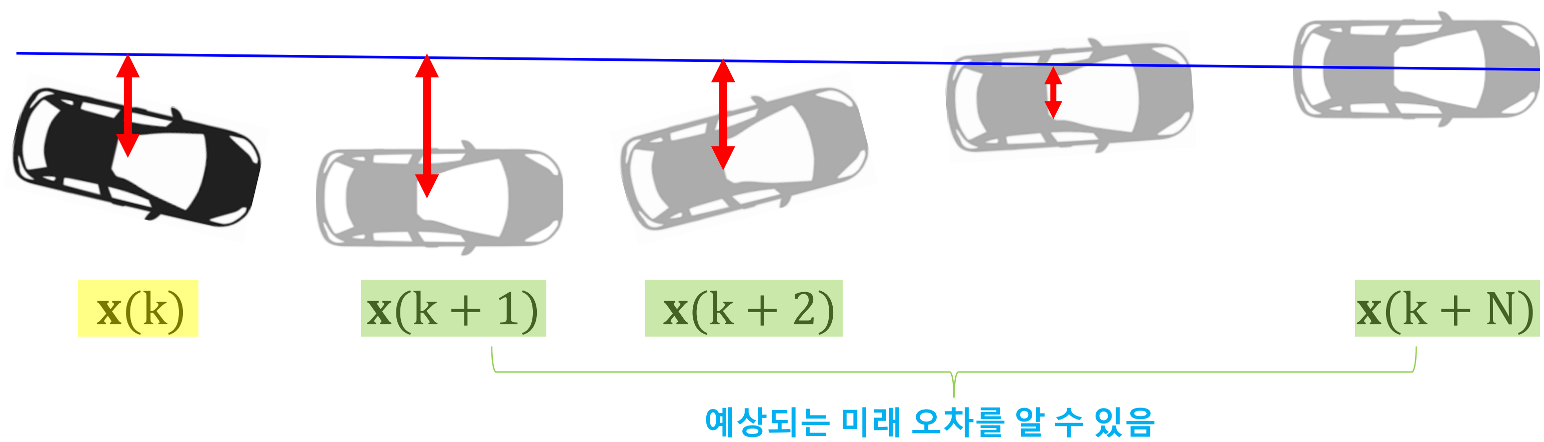


k+N번째의 상태변수와
출력 등을 이산화된 상태공간
모델을 이용하여 예측

$$
\begin{aligned}
\mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) \\
\mathbf{x}(k+2) &= \Phi\mathbf{x}(k+1) + \Gamma\mathbf{u}(k+1) \\
&= \Phi(\Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)) + \Gamma\mathbf{u}(k+1) \\
&= \Phi^2\mathbf{x}(k) + \Phi\Gamma\mathbf{u}(k) + \Gamma\mathbf{u}(k+1) \\
&\vdots \\
\mathbf{x}(k+N) &= \Phi^N\mathbf{x}(k) + \cdots
\end{aligned}
$$

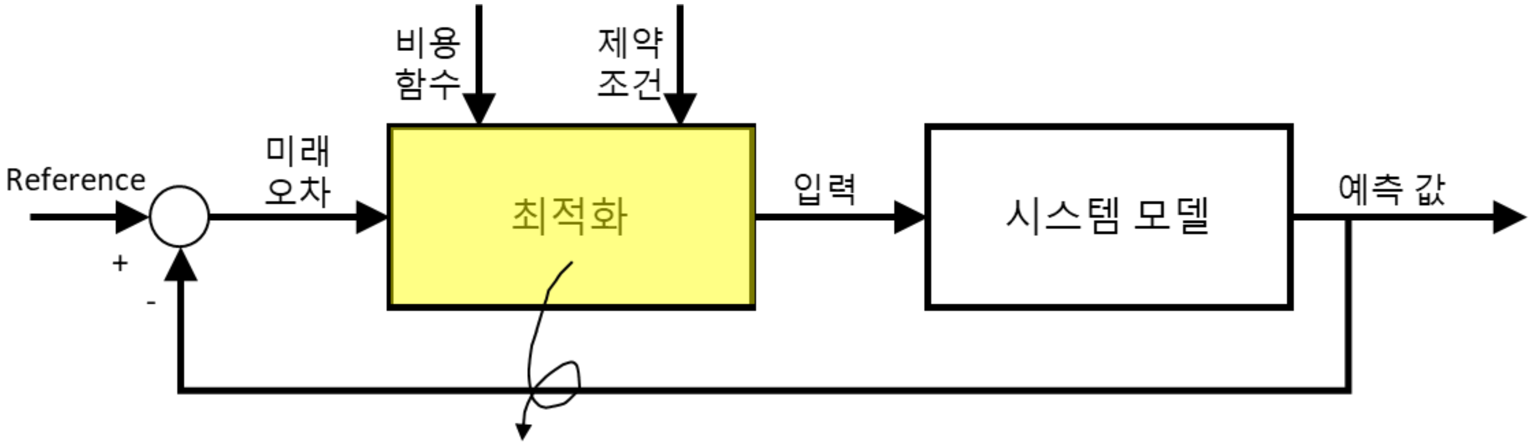
Introduction





k라는 시간에서 예측한 k+i번째의 예측 값 $x(k+i|k)$ $x(k+N|k)$

Introduction



제어의 성능과 제약조건을 동시에
만족하는 최적의 입력값을 만들어야 함

k+N번째까지 어떠한 입력들을
넣어야 조건을 만족

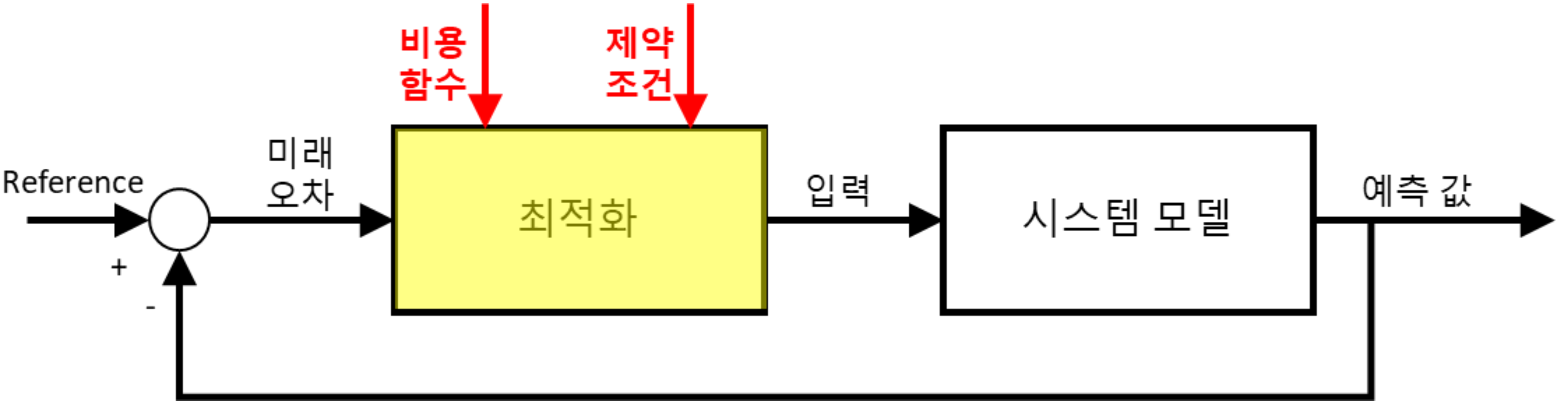
$$U = u(k|k), u(k+1|k), u(k+2|k), \dots, u(k+N|k)$$

현재입력

미래입력

최적의 입력을
찾는 문제

Introduction



k+N번째까지의 상태변수와 입력을 고려하여 설계

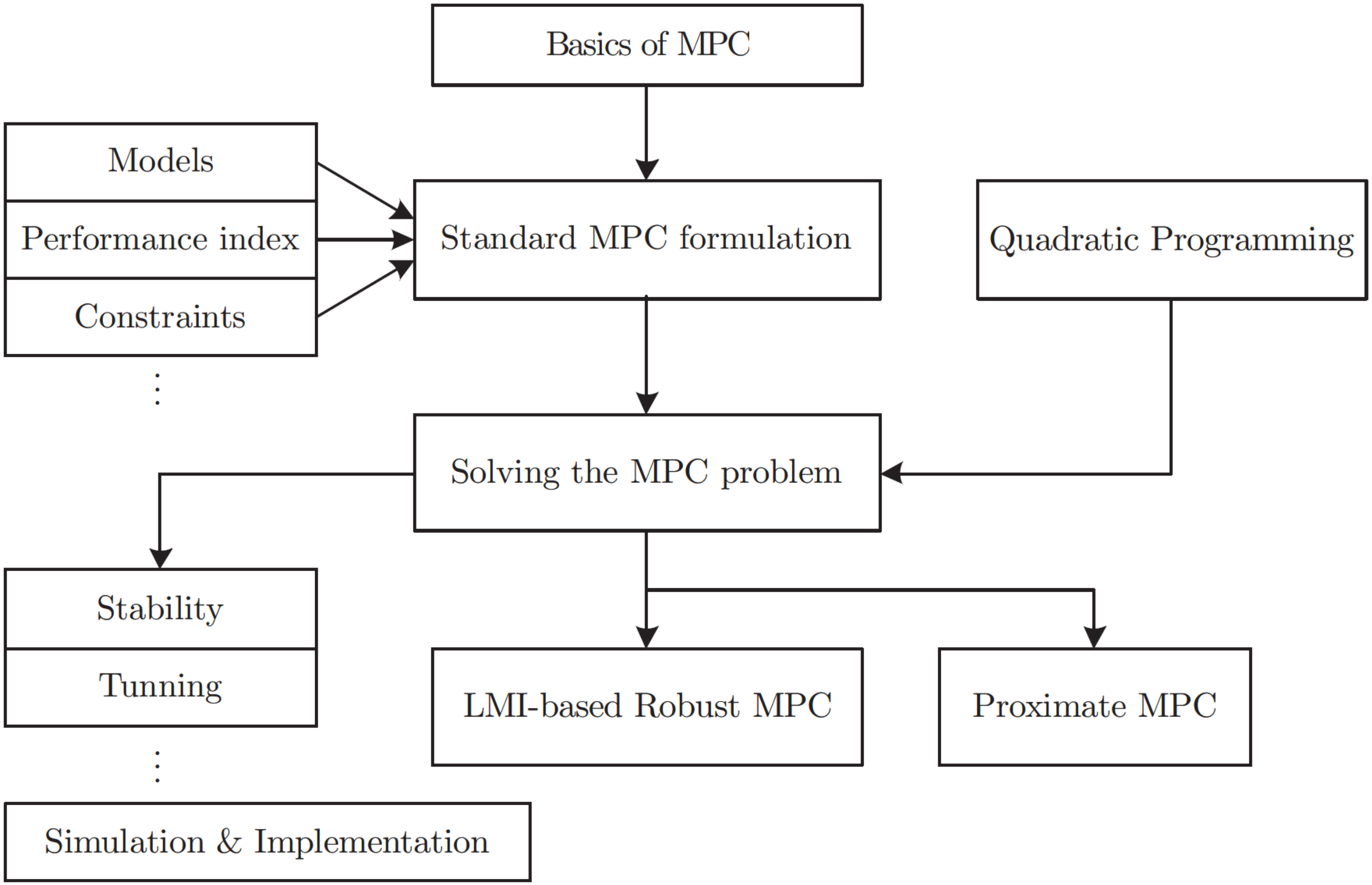
$$J = \sum_{j=k}^{k+N} \underbrace{x^T(j|k)Qx(j|k)}_{\text{상태변수}} + \underbrace{u^T(j|k)Ru(j|k)}_{\text{입력}}$$

$$u_{min} \leq u \leq u_{max}$$
$$x \leq x_{max}$$
$$\vdots$$

최적 입력 U^*

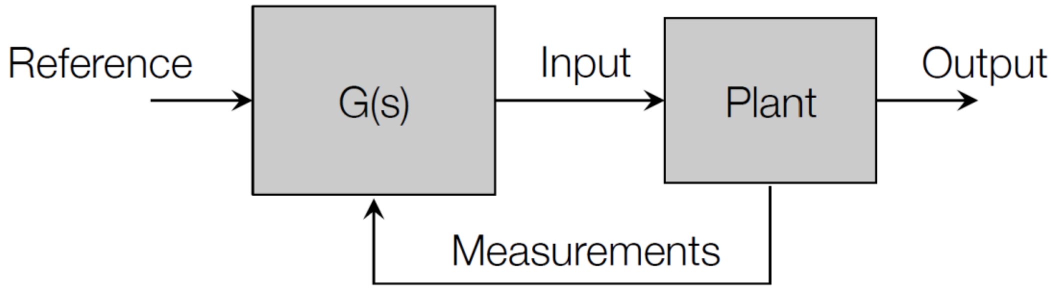
$$\boldsymbol{U^* = u^*(k|k), u^*(k+1|k), u^*(k+2|k), \dots, u^*(k+N|k)}$$

Model predictive control

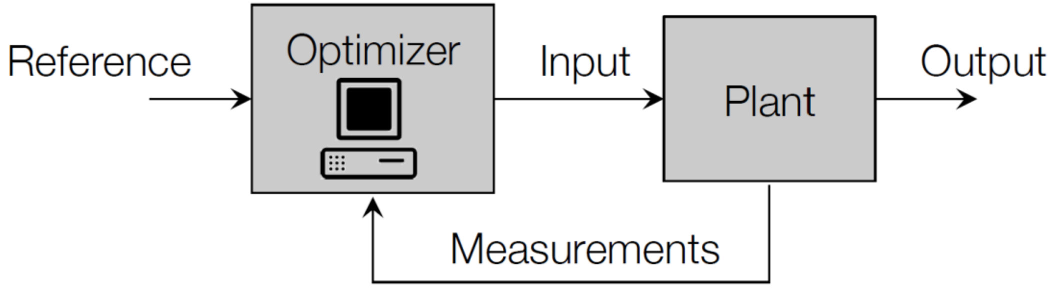


Optimization in the loop

Classical control loop:



The classical controller is replaced by an optimization algorithm:



The optimization uses predictions based on a model of the plant.

Motivation

Objective:

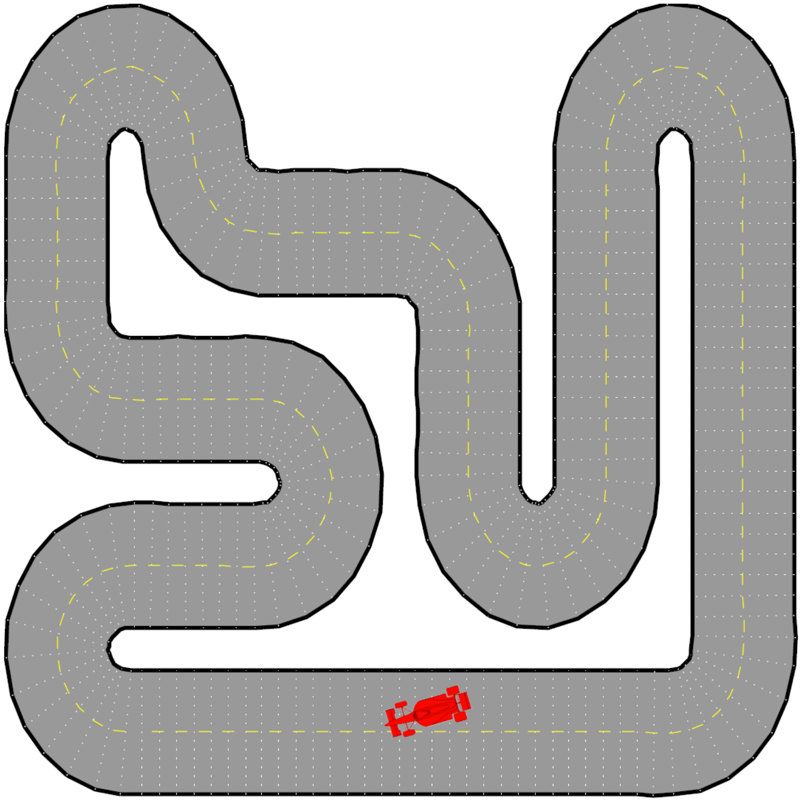
- Minimize lap time

Constraints:

- Avoid other cars
- Stay on road
- Don't skid
- Limited acceleration

Intuitive approach:

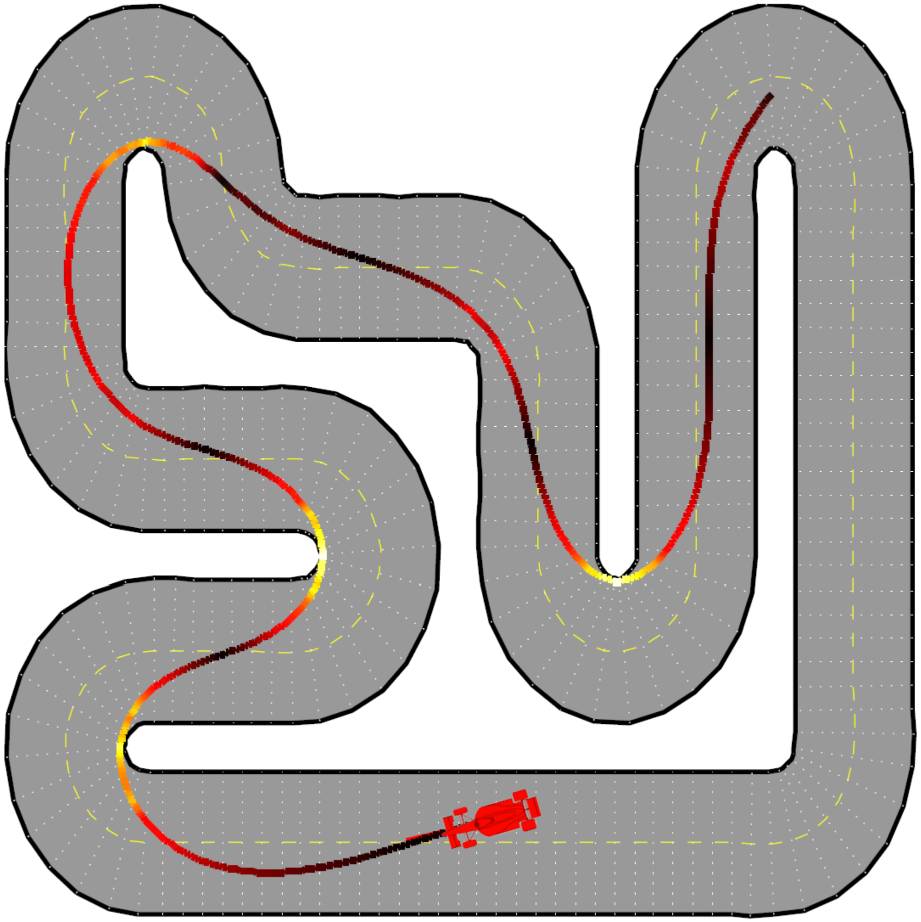
- Look forward and plan path based on
  - Road conditions
  - Upcoming corners
  - Abilities of car
  - etc...



Introduction

Minimize	(lap time)
while	avoid other cars
	stay on road
	…

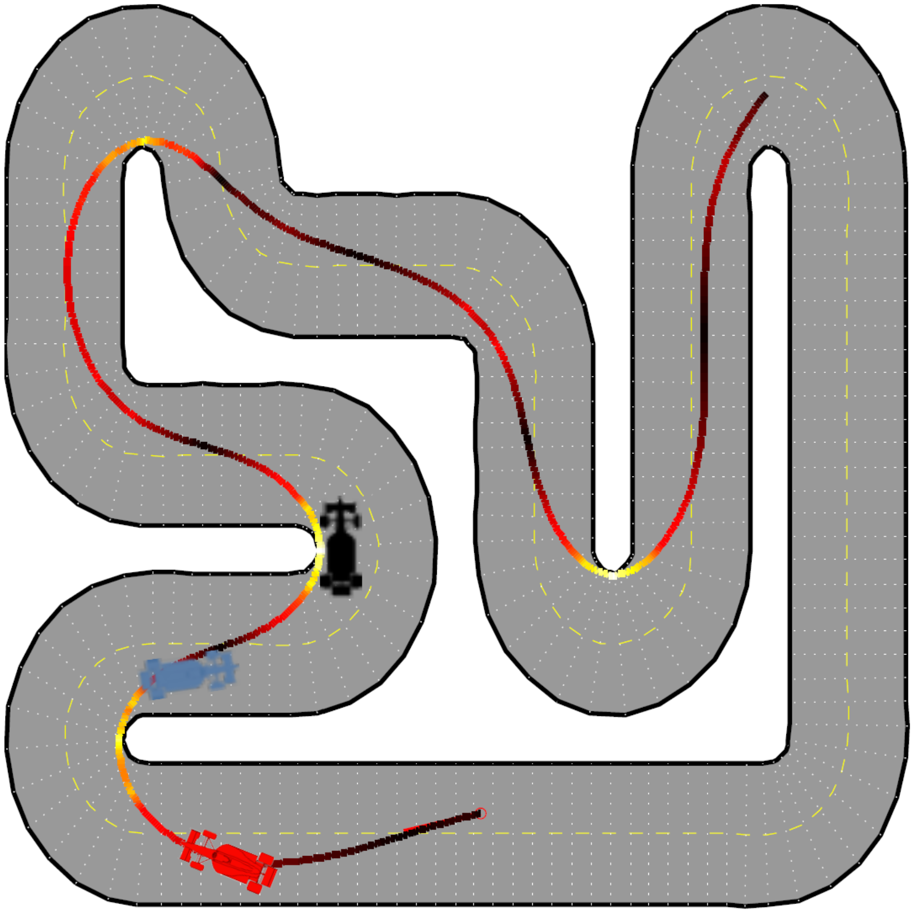
- Solve **optimization problem** to compute minimum-time path



Introduction

Minimize	(lap time)
while	avoid other cars
	stay on road
	...

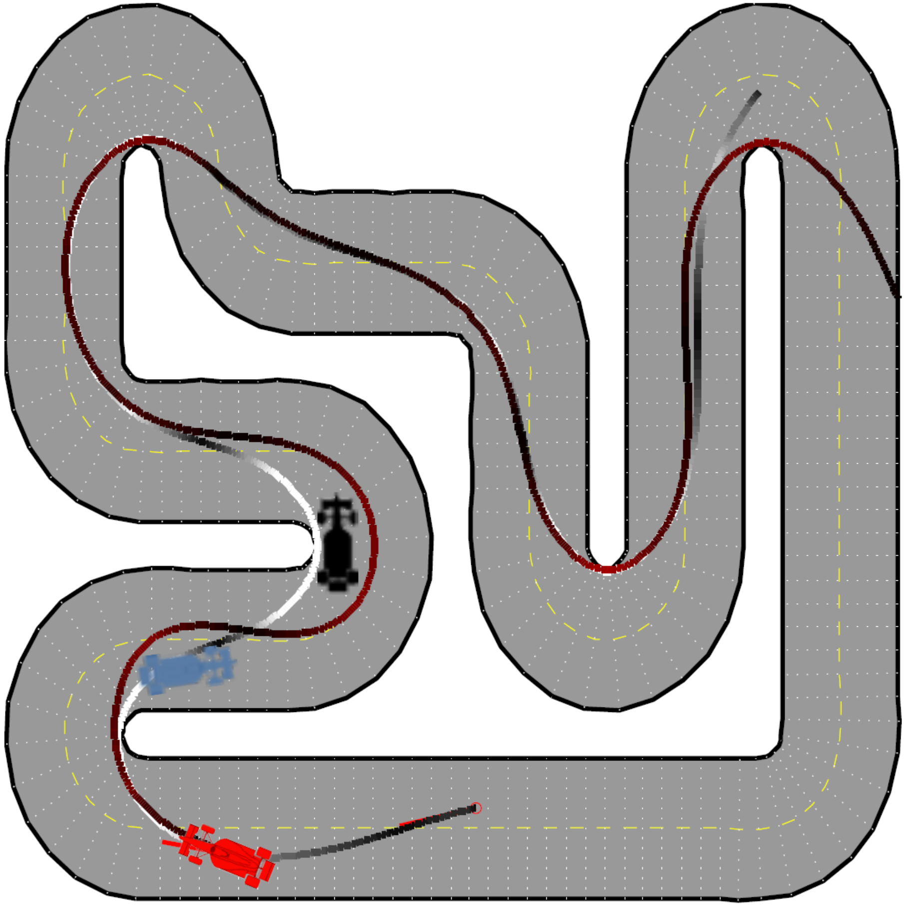
- Solve **optimization problem** to compute minimum-time path
- What to do if something unexpected happens?
  - We didn't see a car around the corner!
  - Must introduce **feedback**



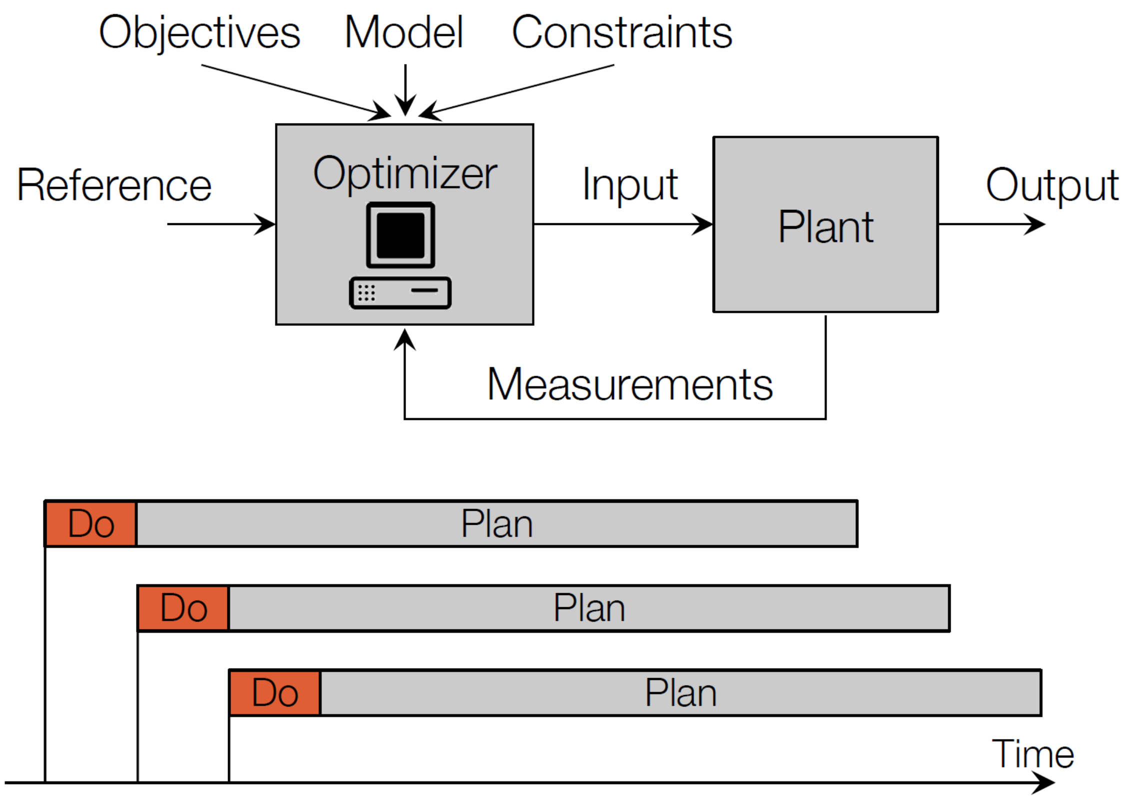
Introduction

Minimize	(lap time)
while	avoid other cars
	stay on road
	...

- Solve **optimization problem** to compute minimum-time path
- Obtain series of planned control actions
- Apply **first** control action
- Repeat the planning procedure



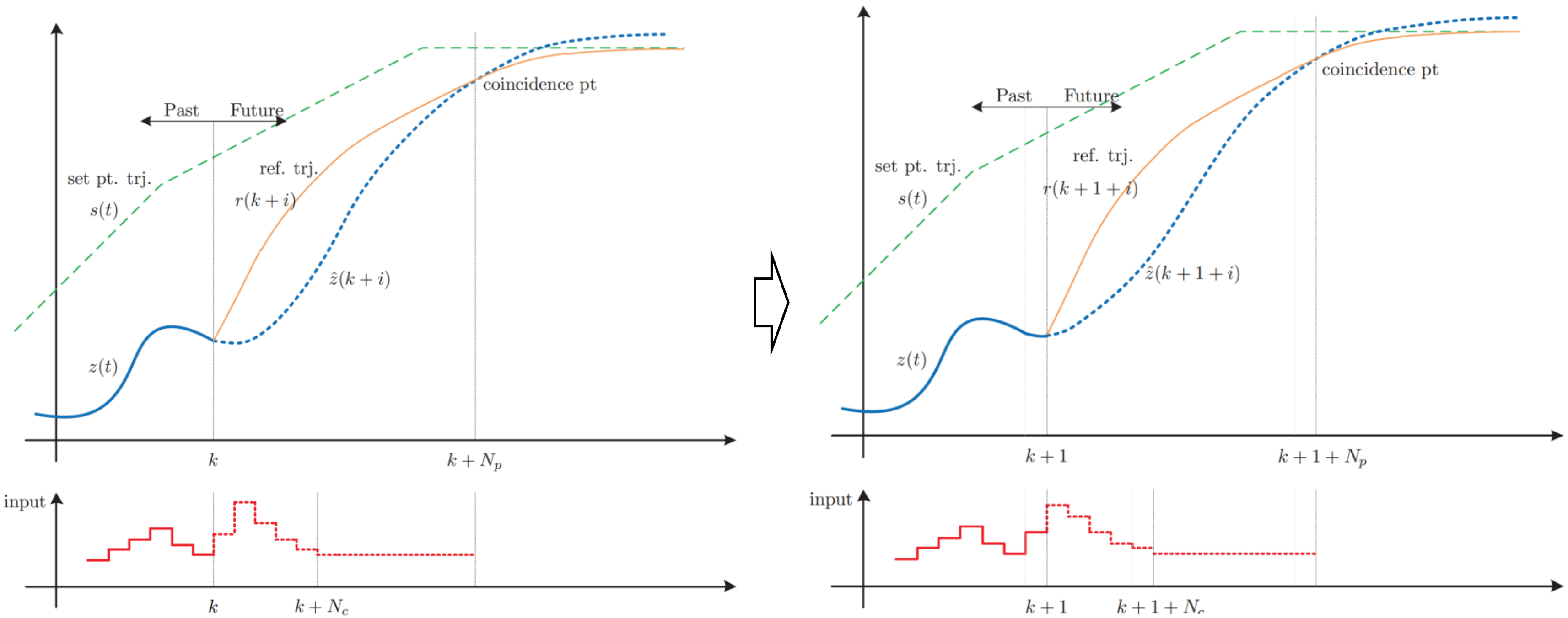
Model Predictive Control



Receding horizon strategy introduces **feedback**.

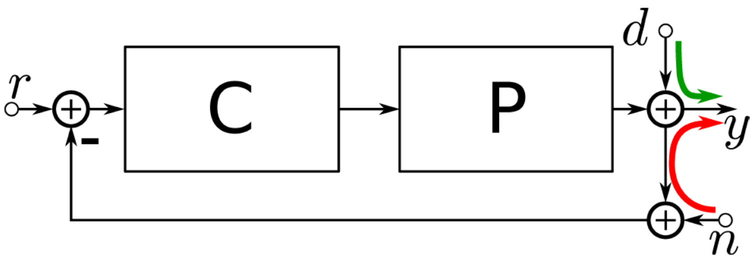
Receding horizon principle

- at time k: Performance index $J(k)$ is minimized
- first element of optimal control sequence $v(k)$ is applied to the system
- horizon shifted
- optimization restarted for time $k+1$



Two Different Perspectives

Classical design: design C

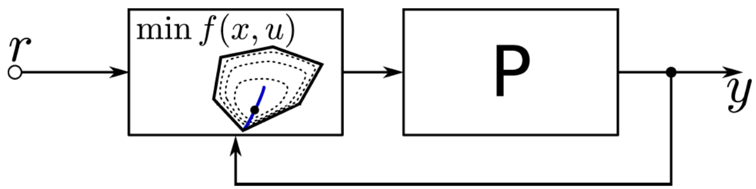


Dominant issues addressed

- Disturbance rejection ($d \to y$)
- Noise insensitivity ($n \to y$)
- Model uncertainty

(usually in **frequency domain**)

MPC: real-time, repeated optimization to choose $u(t)$ – often in supervisory mode



Dominant issues addressed

- Control constraints (limits)
- Process constraints (safety)

(usually in **time domain**)

Constraints in Control

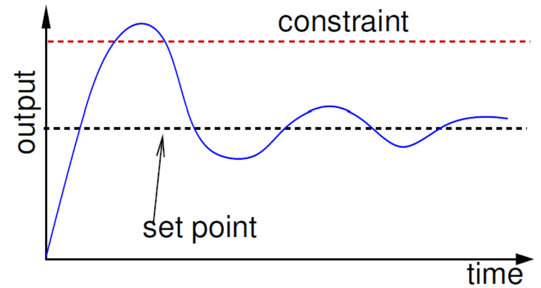
All physical systems have **constraints**:

- Physical constraints, e.g. actuator limits
- Performance constraints, e.g. overshoot
- Safety constraints, e.g. temperature/pressure limits

Optimal operating points are often near constraints.

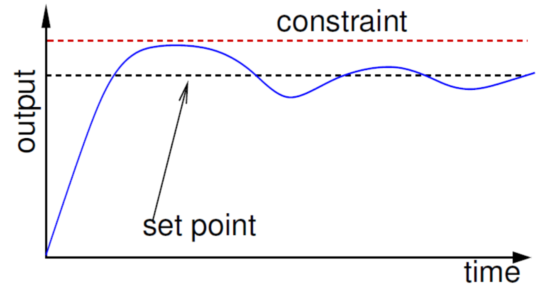
Classical control methods:

- Ad hoc constraint management
- Set point sufficiently far from constraints
- Suboptimal plant operation



Predictive control:

- Constraints included in the design
- Set point optimal
- Optimal plant operation



MPC: Mathematical Formulation

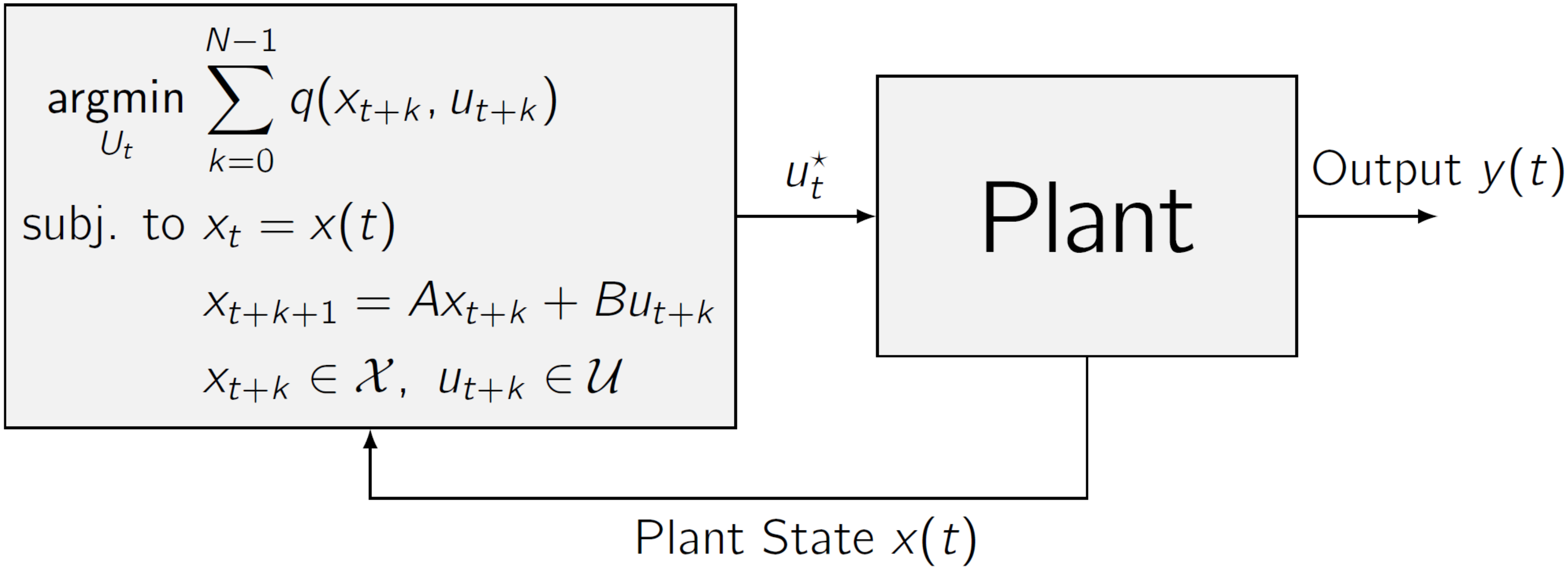
$$U_t^\star(x(t)) := \operatorname*{argmin}_{U_t} \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k})$$

subj. to	$x_t = x(t)$	measurement
	$x_{t+k+1} = Ax_{t+k} + Bu_{t+k}$	system model
	$x_{t+k} \in \mathcal{X}$	state constraints
	$u_{t+k} \in \mathcal{U}$	input constraints
	$U_t = \{u_t, u_{t+1}, \ldots, u_{t+N-1}\}$	optimization variables

Problem is defined by

- **Objective** that is minimized
- Internal **system model** to predict system behavior
- **Constraints** that have to be satisfied

MPC: Mathematical Formulation



At each sample time:

- Measure / estimate current state $x(t)$
- Find the optimal input sequence for the entire planning window N:
  $U_t^\star = \{u_t^\star, u_{t+1}^\star, \ldots, u_{t+N-1}^\star\}$
- Implement only the **first** control action $u_t^\star$

Important Aspects of MPC

Main advantages:

- Systematic approach for handling **constraints**
- High **performance** controller

Main challenges:

- **Implementation**
  MPC problem has to be solved in real-time, i.e. within the sampling interval of the system, and with available hardware (storage, processor,...).
- **Stability**
  Closed-loop stability, i.e. convergence, is not automatically guaranteed
- **Robustness**
  The closed-loop system is not necessarily robust against uncertainties or disturbances
- **Feasibility**
  Optimization problem may become infeasible at some future time step, i.e. there may not exist a plan satisfying all constraints

General Problem Formulation

Consider the nonlinear time-invariant system

$$x(t+1) = g(x(t), u(t))$$

subject to the constraints

$$h(x(t), u(t)) \leq 0, \forall t \geq 0$$

with $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ the state and input vectors. Assume that $g(0,0) = 0$, $h(0,0) \leq 0$.

Consider the following *objective or cost function*

$$J_{0\to N}(x_0, U_{0\to N-1}) := p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k)$$

where

- N is the time *horizon*,
- $x_{k+1} = g(x_k, u_k)$, $k = 0, \dots, N-1$ and $x_0 = x(0)$,
- $U_{0\to N-1} := \left[u_0^\top, \dots, u_{N-1}^\top\right]^\top \in \mathbb{R}^s$, $s = mN$,
- $q(x_k, u_k)$ and $p(x_N)$ are the *stage cost* and *terminal cost*, respectively.

Objectives

- **Finite Time Solution**
  - a general nonlinear programming problem (*batch approach*)
  - recursively by invoking Bellman's Principle of Optimality (*recursive approach*)
  - discuss in details the linear system case
- **Infinite Time Solution**. We will investigate
  - if a solution exists as $N \to \infty$
  - the properties of this solution
  - approximate of the solution by using a *receding horizon technique*
- **Uncertainty**. We will discuss how to extend the problem description and consider uncertianty.

Linear Quadratic Optimal Control

- In this section, only **linear** discrete-time time-invariant systems

$$x(k+1) = Ax(k) + Bu(k)$$

  and **quadratic** cost functions

$$J_0(x_0, U) := x_N^\top P x_N + \sum_{k=0}^{N-1} (x_k^\top Q x_k + u_k^\top R u_k) \qquad (1)$$

  are considered, and we consider only the problem of regulating the state to the origin, **without state or input constraints**.
- The two most common solution approaches will be described here
  1. **Batch Approach**, which yields a series of **numerical values** for the input
  2. **Recursive Approach**, which uses Dynamic Programming to compute control **policies** or **laws**, i.e. functions that describe how the control decisions depend on the system states.

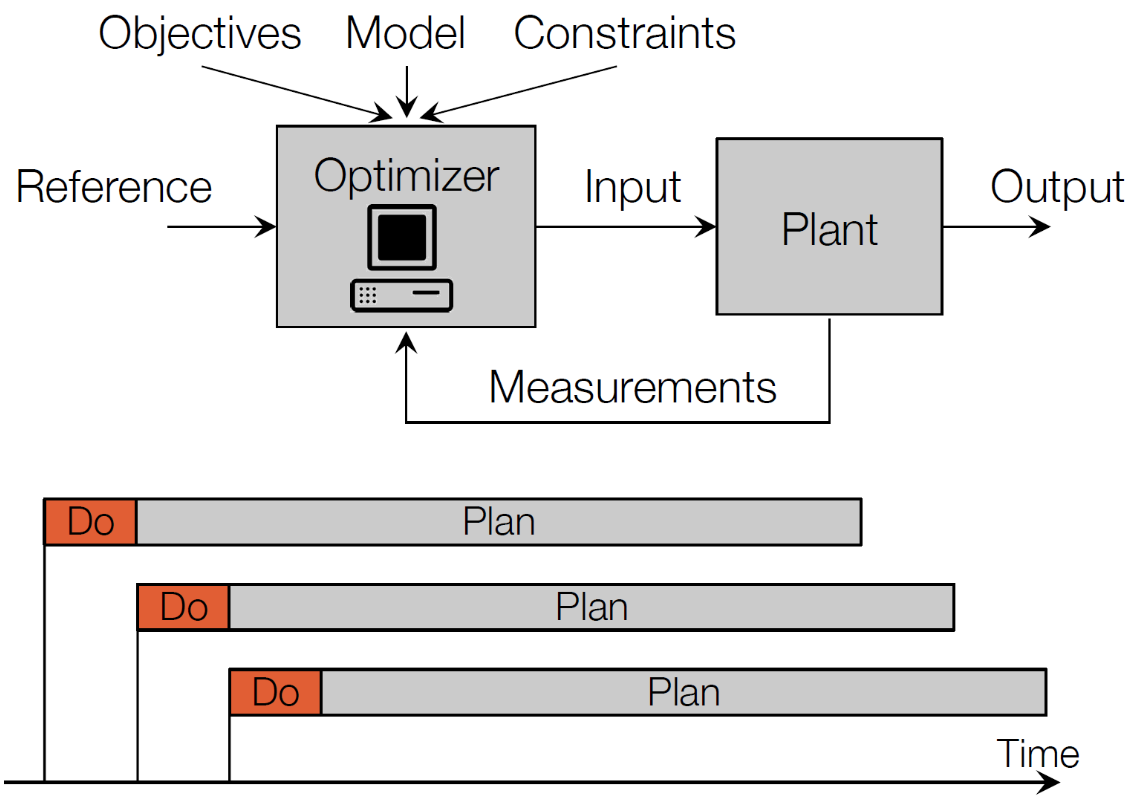
Unconstrained Finite Horizon Control Problem

- **Goal:** Find a sequence of inputs $U_{0\to N-1} := [u_0^\top, \dots, u_{N-1}^\top]^\top$ that minimizes the objective function

$$
\begin{aligned}
J_0^\star(x(0)) := \min_{U_{0\to N-1}} \; & x_N^\top P x_N + \sum_{k=0}^{N-1} (x_k^\top Q x_k + u_k^\top R u_k) \\
\text{subj. to } & x_{k+1} = A x_k + B u_k, \; k = 0, \dots, N-1 \\
& x_0 = x(0)
\end{aligned}
$$

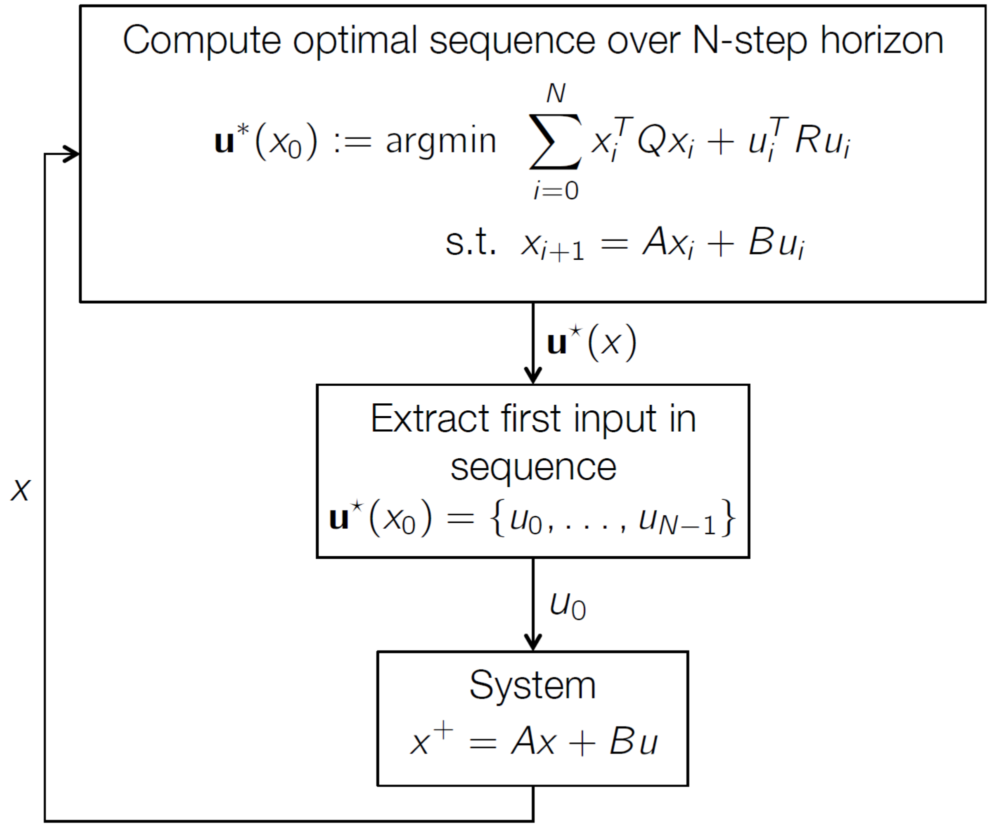
- $P \succeq 0$, with $P = P^\top$, is the **terminal** weight
- $Q \succeq 0$, with $Q = Q^\top$, is the **state** weight
- $R \succ 0$, with $R = R^\top$, is the **input** weight
- N is the horizon length
- Note that $x(0)$ is the current state, whereas $x_0, \dots, x_N$ and $u_0, \dots, u_{N-1}$ are **optimization variables** that are constrained to obey the system dynamics and the initial condition.

Receding Horizon Control



Receding horizon strategy introduces feedback.

Receding Horizon Control



For unconstrained systems, this is a **constant linear controller**
However, can extend this concept to much more complex systems (MPC)

Example - Impact of Horizon Length

Consider the lightly damped, stable system

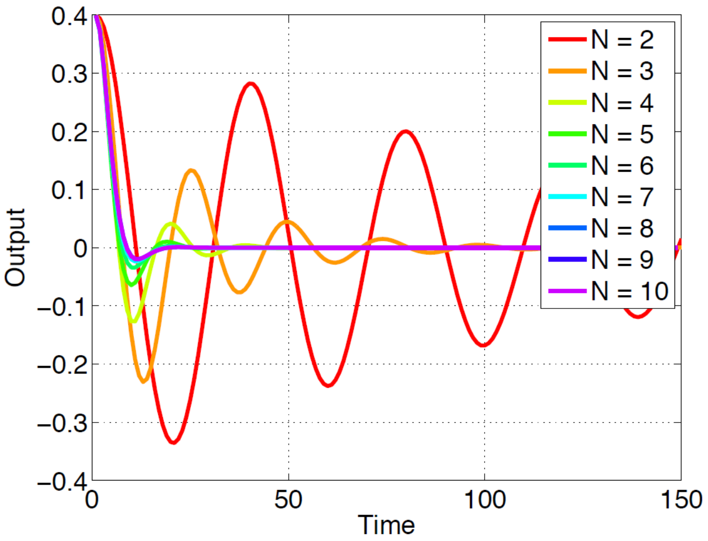
$$G(s) := \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

where $\omega = 1$, $\zeta = 0.01$. We sample at 10Hz and set $P = Q = I$, $R = 1$.

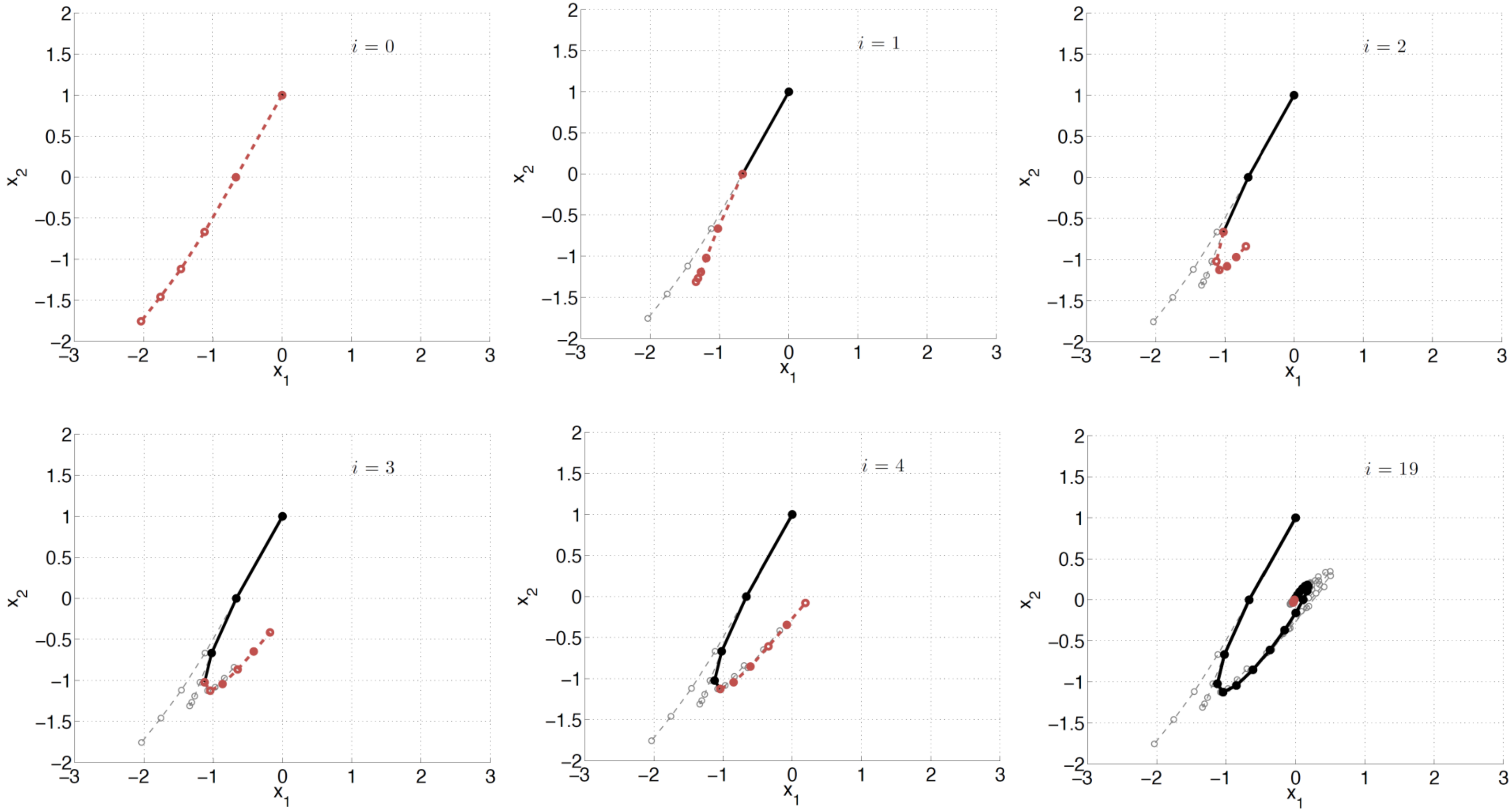
Discrete-time state-space model:

$$x^+ = \begin{bmatrix} 1.988 & -0.998 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0.125 \\ 0 \end{bmatrix} u$$

Closed-loop response

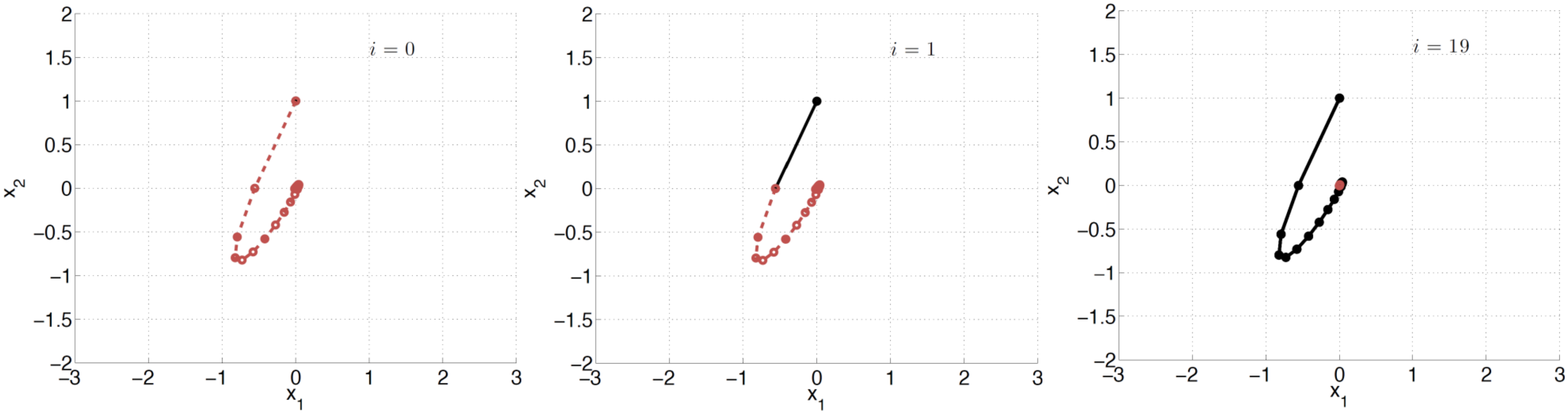


Example: Short horizon N = 5



Short horizon: Prediction and closed-loop response differ.

Example: Long horizon N = 20



Long horizon: Prediction and closed-loop match.

Stability of Finite-Horizon Optimal Control Laws

Consider the system

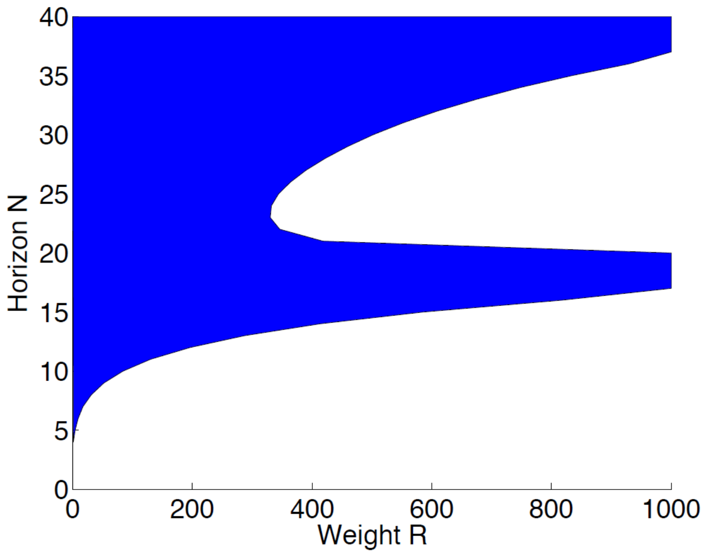
$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

where $\omega = 0.1$ and $\zeta = -1$, which has been discretized at $1r/s$.
(Note that this system is unstable)

Is the system $x^+ = (A + BK_{R,N})x$ stable?

Where $K_{R,N}$ is the finite horizon LQR controller with horizon N and weight R (Q taken to be the identity)

Blue = stable, white = unstable



Infinite Horizon Control Problem: Optimal Solution

- In some cases we may want to solve the same problem with an infinite horizon:

$$J_\infty(x(0)) = \min_{u(\cdot)} \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right)$$

$$\text{subj. to } x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, 2, \ldots, \infty,$$

$$x_0 = x(0)$$

- As with the Dynamic Programming approach, the optimal input is of the form

$$u^\star(k) = -(B^\top P_\infty B + R)^{-1} B^\top P_\infty A x(k) := F_\infty x(k)$$

and the infinite-horizon cost-to-go is

$$J_\infty(x(k)) = x(k)^\top P_\infty x(k) .$$

Infinite Horizon Control Problem: Optimal Solution

- The matrix P_∞ comes from an infinite recursion of the RDE, from a notional point infinitely far into the future.
- Assuming the RDE does converge to some constant matrix P_∞, it must satisfy the following (from (6), with $P_k = P_{k+1} = P_\infty$)

$$P_\infty = A^\top P_\infty A + Q - A^\top P_\infty B(B^\top P_\infty B + R)^{-1} B^\top P_\infty A,$$

which is called the **Algebraic Riccati equation (ARE)**.

- The constant feedback matrix F_∞ is referred to as the asymptotic form of the **Linear Quadratic Regulator (LQR)**.
- In fact, if (A, B) is stabilizable and $(Q^{1/2}, A)$ is detectable, then the RDE (initialized with Q at $k = \infty$ and solved for $k \searrow 0$) converges to the unique positive definite solution P_∞ of the ARE.

Summary