

# ESE 619 Model Predictive Control

## Lecture 0: Course Logistics

### I. Lecture Material and Schedule

- **Compilation:** Manfred Morari and Melanie Zeilinger
- **Book:** Predictive Control for Linear and Hybrid Systems by F. Borrelli, A. Bemporad and M. Morari Cambridge University Press, 2017
- **Grades:** 30% Problem Sets 25% Midterm Exam 45% Final Exam

#### Schedule:

Chapter 1: Introduction and Overview

Chapter 2: System Theory Basics

Chapter 3: Uncertainty and State Estimation

Chapter 4: Convex Optimization

Chapter 5: Unconstrained Linear Quadratic

Optimal Control

Chapter 6: Constrained Finite Time Optimal

Control

Chapter 7: Feasibility and Stability

Chapter 8: Invariance

Chapter 9: Reachability and Invariant Sets

Chapter 10: Practical Issues

Chapter 11: Explicit MPC

Chapter 12: Hybrid MPC

Chapter 13: Robust MPC

Chapter 14: Numerical Methods

Chapter 14/15: Numerical Methods Operator

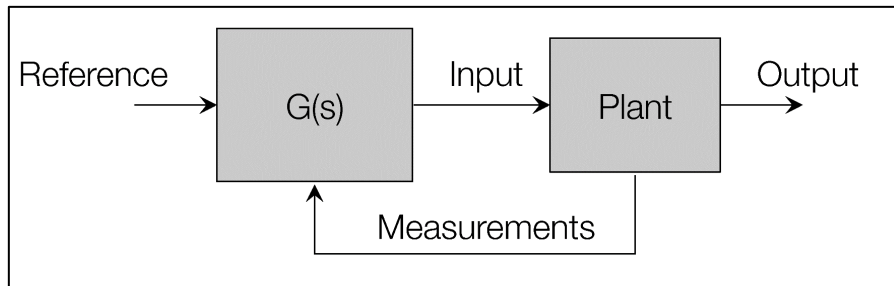
Splitting Methods

# Lecture 1: Introduction and MPC Overview

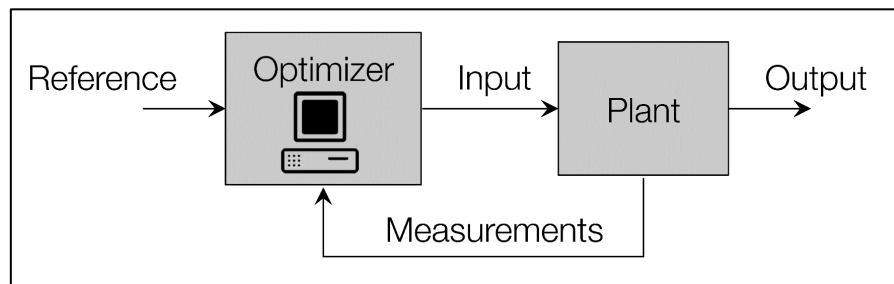
## I. Optimization-based Control

- **Optimization in the loop**

Classical (like PID or PI) control loop (plant means the real system):



The classical controller is replaced by an optimization algorithm (much more complex controller):



The optimization uses predictions based on a model of the plant **due to the progress of hardware**.

**Example (Motivation):** race car control

Objective: Minimize lap time

Constraints: Avoid other cars, Stay on road, Don't skid, Limited acceleration

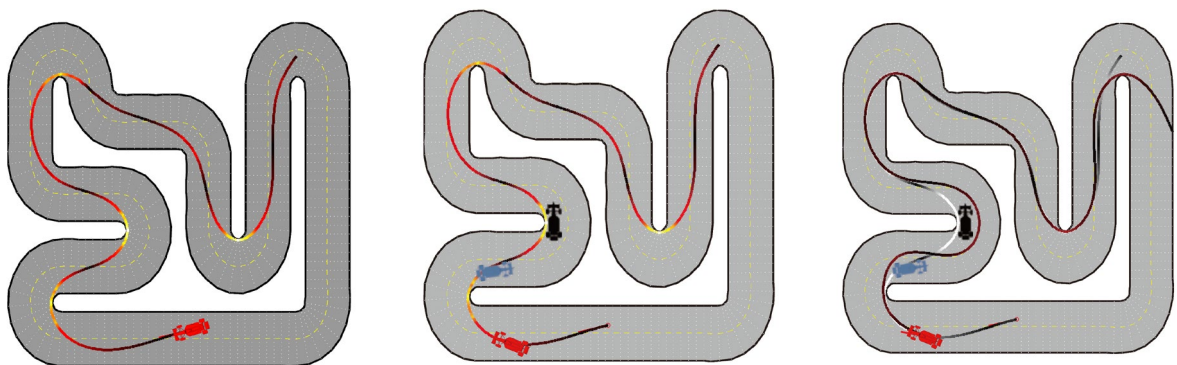
Intuitive approach: Look forward and plan path based on: Road conditions, Upcoming corners etc.

Solve optimization problems to compute the minimum-time path. But the result is fragile.

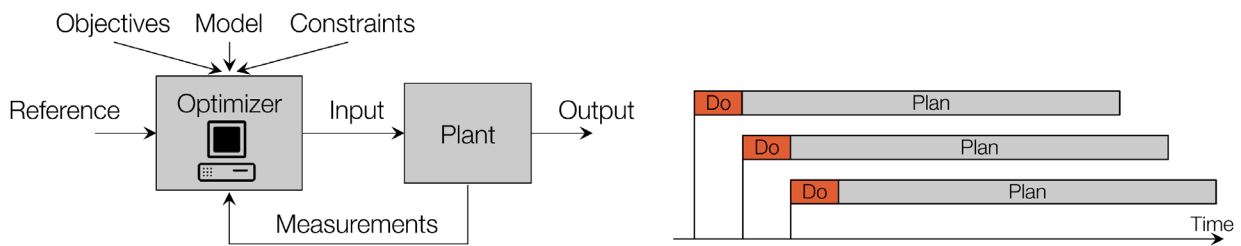
⇒ Must introduce **feedback**

⇒ Obtain a series of planned control actions:

⇒ Apply the first control action, repeat the planning procedure



- **MPC Workflow**

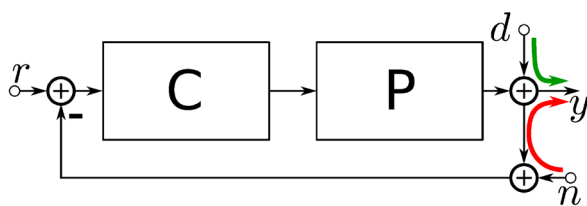


**Receding (Moving) horizon** strategy introduces **feedback**.

## II. Concepts of MPC

- **Two Different Perspective for Controller Design**

**Classical control design:** design control  $C$ : usually done in **frequency domain** (low-level)



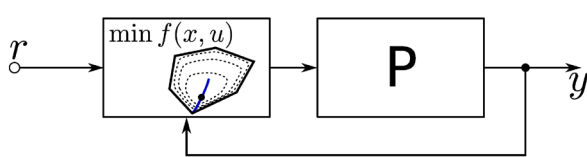
Dominant issues (control objective) for feedback:

Disturbance rejection ( $d \rightarrow y$ )

Noise insensitivity ( $n \rightarrow y$ )

Model uncertainty. (Trade-off between the two)

**MPC control design:** real-time, repeated optimization to choose  $u(t)$ , in **time domain** (high-level)



Dominant issues (control objective) for feedback:

Control constraints (limits)

Process constraints (safety)

**In a supervisory mode (based on classical control)**

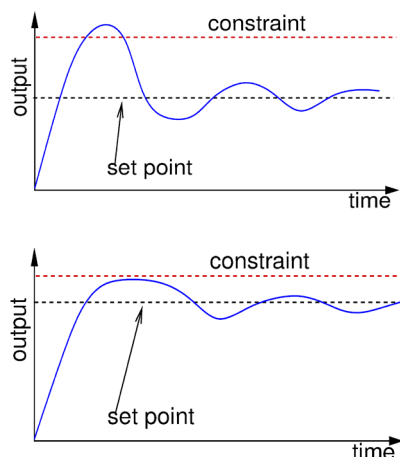
**Note:** These two are not against each other, generally they are used together (high and low level control)

- **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

**Key point:** 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

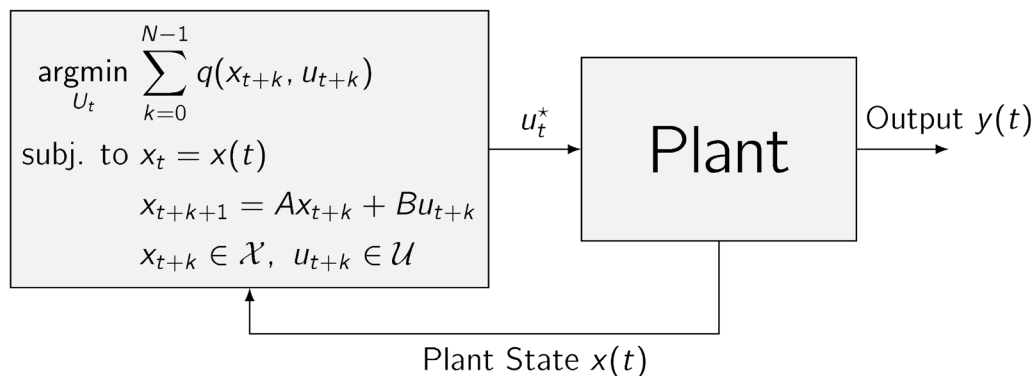
Set point optimal

Optimal plant operation

- **MPC: Mathematical Formulation**

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

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  
 $U_t = \{u_t, u_{t+1} \dots, u_{t+N-1}\}$  optimization variables



Problem is defined by:

1. **Objective** that is minimized
2. Internal **system model** to predict system behavior
3. **Constraints** that have to be satisfied

At each sample time:

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

Note: Always remember that MPC is a **feedback** controller. However, also remember that it is a very different feedback controller, it can also suffer from stability problems, e.g. if you take a very short-sighted horizon, in the long-term horizon sense, it could lead to terrible behavior.

### III. Applications of MPC (Referred to Slides)

Ball on Plate

Kite Power

Path Following

Automotive Systems

Autonomous Quadcopter Flight

Catalytic Cracker

Autonomous dNaNo Race Cars

Predictive Control in NeuroScience

Energy Efficient Building Control

## IV. History of MPC

A. I. Propoi, 1963, “Use of linear programming methods for synthesizing sampled-data automatic systems”, Automation and Remote Control.

J. Richalet et al., 1978 “Model predictive heuristic control- application to industrial processes”. Automatica, 14:413-428.

1970s: Cutler suggested MPC in his PhD proposal at the University of Houston in 1969 and introduced it later at Shell under the name Dynamic Matrix Control. C. R. Cutler, B. L. Ramaker, 1979 “Dynamic matrix control - a computer control algorithm”. AICHE National Meeting, Houston, TX.

C. Cutler, A. Morshedi, J. Haydel, 1983. “An industrial perspective on advanced control”. AICHE Annual Meeting, Washington, DC.

Mid 1990s: extensive theoretical effort devoted to provide conditions for guaranteeing feasibility and closed-loop stability

2000s: development of tractable robust MPC approaches; nonlinear and hybrid MPC; MPC for very fast systems.

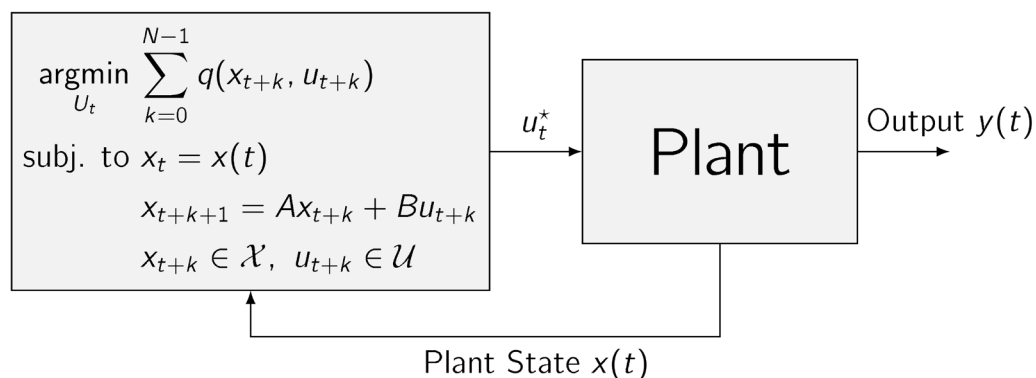
2010s: stochastic MPC; distributed large-scale MPC; economic MPC

## V. Summary

- **MPC Core Algorithm:**

At each sample time:

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



- **MPC Workflow:**

Obtain a model of the system

Design a state observer

Define optimal control problem

Set up optimization problem in optimization software

Solve optimization problem to get optimal control sequence

Verify that closed-loop system performs as desired,

e.g., check performance criteria, robustness, real-time aspects,...

- **Advantages of MPC:**

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