ME 290J
Model Predictive Control
for Linear and Hybrid Systems

Francesco Borrelli

Spring 2011

Department of Mechanical Engineering

University of California

Berkeley, USA

Instruction

- **Instructor:** Francesco Borrelli, Room 5139 EH, 643-3871, fborrelli@berkeley.edu
  Office Hours: Tu and Th 3.30-4.30
- **Teaching Assistant:** None – might change.
- **Lectures:** Tu-Th 2-3.30 Room 81, Evans Hall
- **Class Notes:** Slides distributed before (sometime after) the class
- **Class Web Site:** bSpace

Matlab

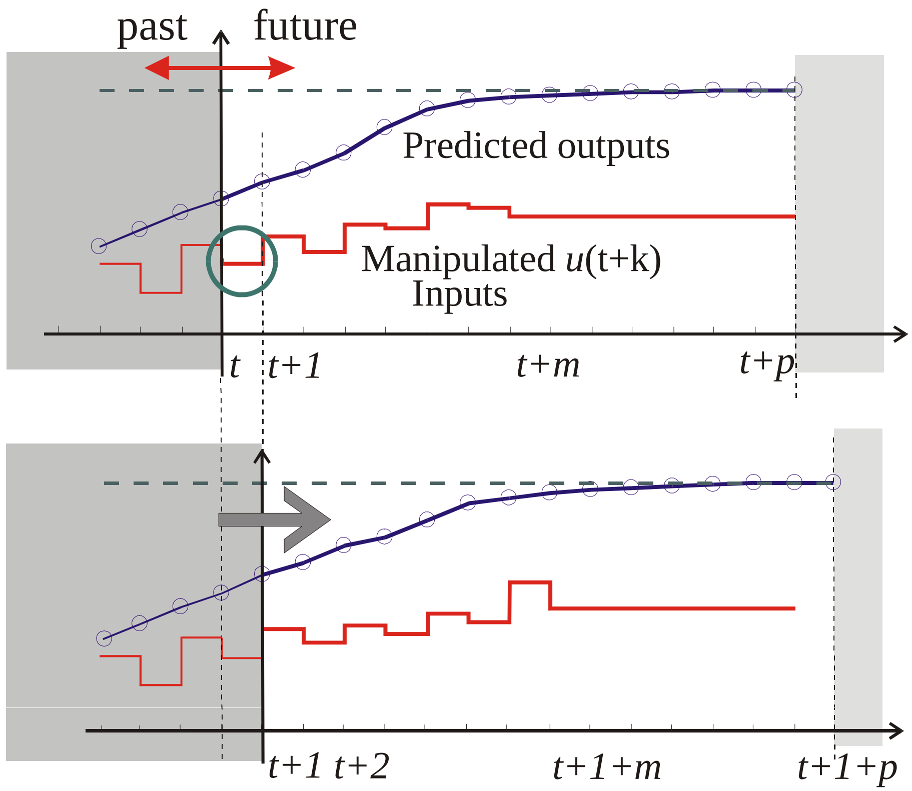
- Matlab running the computers in 2109 Etcheverry Hall
- Card key access required
- I will submit class list to so that everyone in the class has access to that room
  - Please enroll in the class ASAP
- **Need** additional toolbox/Software distributed through bSpace

Grading

- 20% Homework
- 50% Final Project
- 30% Oral or Final Test

ME290J Overview

Model Predictive Control



- Optimize at time *t* (new measurements)
- Only apply the first optimal move *u(t)*
- Repeat the whole optimization at time *t +1*
- Optimization using current measurements □ Feedback

MPC Algorithm

$$\min_{U} \sum_{k=t}^{t+N-1} l(x_k, u_k)$$

$$\text{subj. to} \begin{cases} x_{k+1} = f(x_k, u_k), & k = t, \dots, t+N-1 \\ u_k \in \mathcal{U}, & k = t, \dots, t+N-1 \\ x_k \in \mathcal{X}, & k = t, \dots, t+N-1 \\ x_t = x(t) \end{cases}$$

<u>At time t:</u>

- Measure (or estimate) the current state ***x(t)***
- Find the optimal input sequence $U^* = \{u^*_t, u^*_{t+1}, u^*_{t+2}, \dots, u^*_{t+N-1}\}$
- Apply only $u(t)=u^*_t$, and discard $u^*_{t+1}, u^*_{t+2}, \dots$

<u>Repeat the same procedure at time *t +1*</u>

Multivariable, Model Based

Nonlinear, Constraints Satisfaction, Prediction

Important Issues in Model Predictive Control

Even assuming perfect model, no disturbances:

$$\boxed{\begin{array}{c}\text{predicted open-loop trajectories}\\ \neq \\ \text{closed-loop trajectories}\end{array}}$$

- **Feasibility**
  Optimization problem may become infeasible at some future time step.

- **Stability**
  Closed-loop stability is not guaranteed.

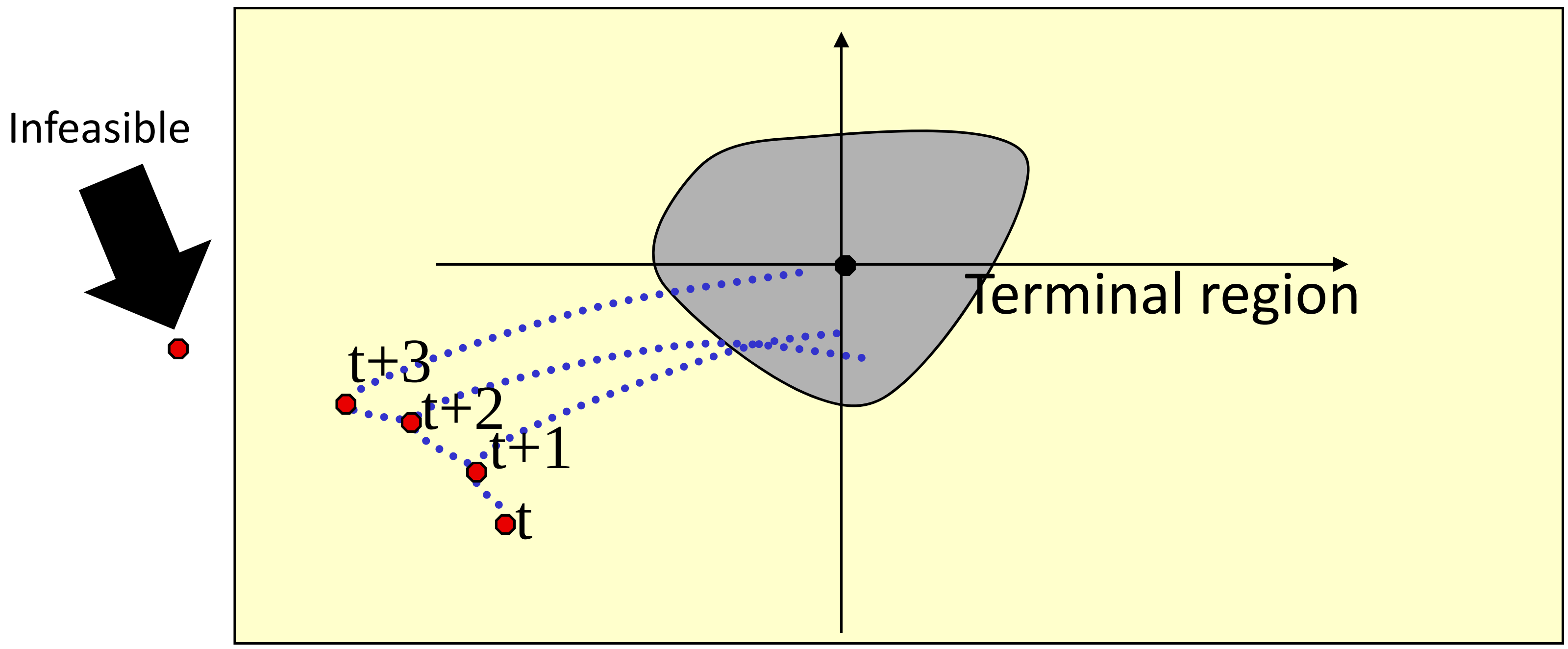
- **Performance**

  Goal: $\min \sum_{k=t}^{\infty} l(x_k, u_k)$

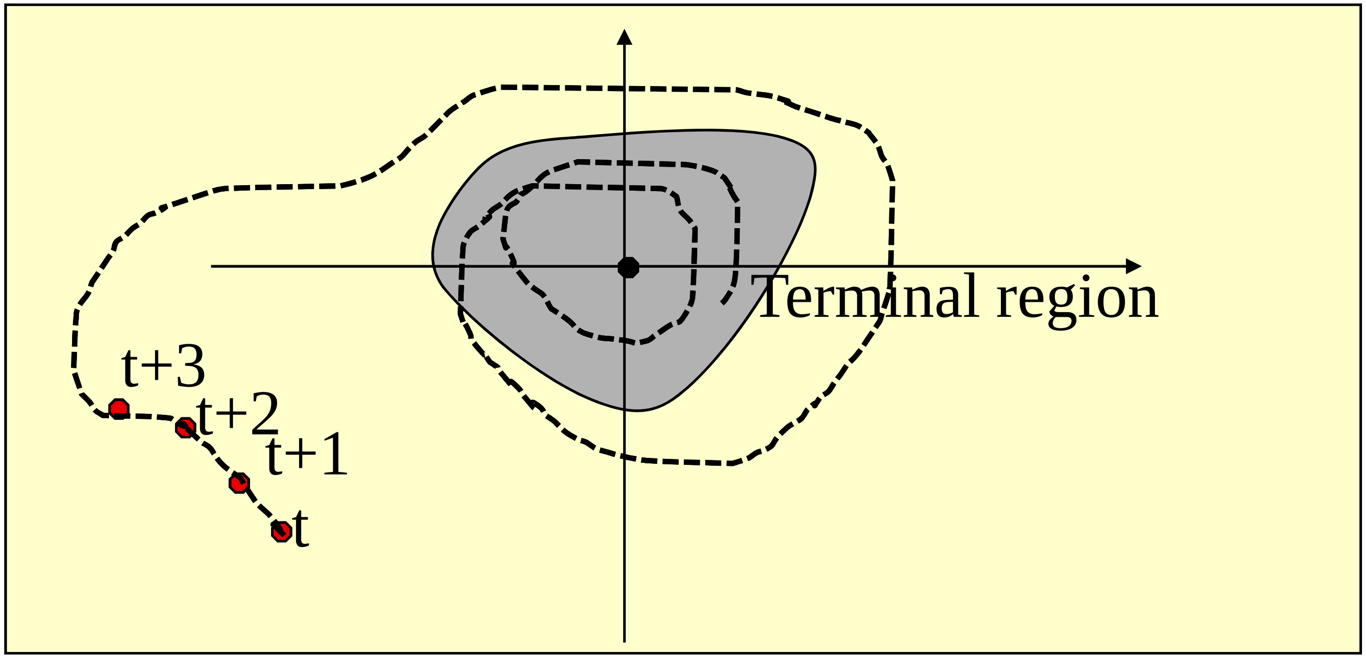
  What is achieved by repeatedly minimizing $\min \sum_{k=t}^{t+N} l(x_k, u_k)$

- **Real-Time Implementation**

Feasibility Issues



Stability Issues



Feasibility and Stability Constraints

$$
\begin{aligned}
\min_{U} \quad & \sum_{k=t}^{t+N-1} l(x_k, u_k) + p(x_{t+N}) \\
\text{subj. to} \quad & \begin{cases} x_{k+1} = f(x_k, u_k), \; k = t, \ldots, t+N-1 \\ u_k \in \mathcal{U}, \; k = t, \ldots, t+N-1 \\ x_k \in \mathcal{X}, \; k = t, \ldots, t+N-1 \\ x_{t+N} \in \mathcal{X}_f \\ x_t = x(t) \end{cases}
\end{aligned}
$$

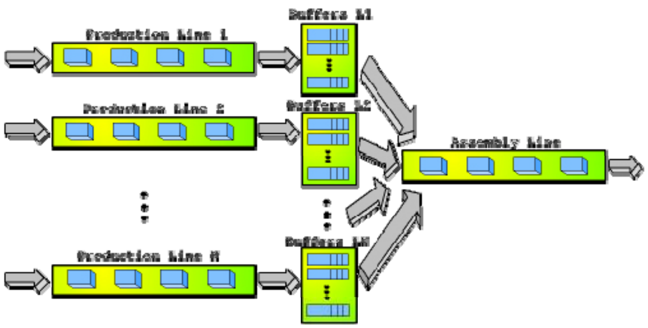
Modified Problem

(Large Body of Literature)

X_f ***(Robust) Invariant Set***

p(x) Control Lyapunov Function

Real Time Implementation



Summarizing…

Need:

- A discrete-time model of the system

  (Matlab, Simulink)

- A state observer
- Set up an Optimization Problem

  (Matlab, MPT toolbox/Yalmip)

- Solve an optimization problem

  (Matlab/Optimization Toolbox, NPSOL, NAG, Cplex)

- Verify that the closed-loop system performs as desired (avoid infeasibility/stability)
- Make sure it runs in real-time and code/download for the embedded platform

Class Topics

(Subject to changes)

Week 1: Introduction and Fundamentals of Optimization

Week 2/3: Fundamentals of Optimization.

Week 4: Multiparametric Programming.

Week 5. Convex and Mixed-Integer Multiparametric Programming

Week 6: Review of Optimal Control Theory and Dynamic Programming

Week 7: Invariant Set Theory

Week 8/9: Constrained Optimal Control for Linear Systems

Week 10: Predictive Control: Fundamentals

Week 11: Predictive Control: Stability and Feasibility Theory

Week 12/13: Hybrid systems

Week 14: Predictive Control for Hybrid Systems.

Week 15/16: Applications/Case Studies

Class Goals

- Predictive Control Theory
- Design and Implement an MPC Controller in Matlab
- Computational Oriented Models of Hybrid Systems

Initial Remarks

- **MPC Name**
- Continuous-Time versus Discrete-Time
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- Non trivial examples
- What I'd like to add:

  Distributed, Robust, Probabilistic, Soft-Constraints

- Any preference for examples?

Initial Remarks

- MPC Name
- **Continuous-Time versus Discrete-Time**
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- Non trivial examples
- What I'd like to add:

  Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

Why Not Continuous Time Optimization?

$$
\begin{array}{lll}
J^*(x(t)) = & \min_{u_{[t:t-N]}} & \int_t^{t-N} l(x(\tau), u(\tau))d\tau \\
& \text{subj.to} & \dot{x}(\tau) = f(x(\tau), u(\tau)), \quad \tau \in [t, t+N] \\
& & x(\tau) \in \mathcal{X}, \quad \tau \in [t, t+N] \\
& & u(\tau) \in \mathcal{U}, \quad \tau \in [t, t+N]
\end{array}
$$

- **Choice of this course**
- **Issues are the same**
- **Soon or later need to discretize**
- **Software exists (Example: http://tomdyn.com/)**

Initial Remarks

- MPC Name
- Continuous-Time versus Discrete-Time
- **MPC in Practice**
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- Non trivial examples
- What I'd like to add: Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

MPC in Practice

- Second Most Used Control Methodology after PID

Qin, S. J. and T. A. *Bagdwell* (2003). A survey of model *predictive control* tecnology. Control Engineering Practice 11, 733-764

- Persistent Control Community Interest (see ACC, CDC proceeding)

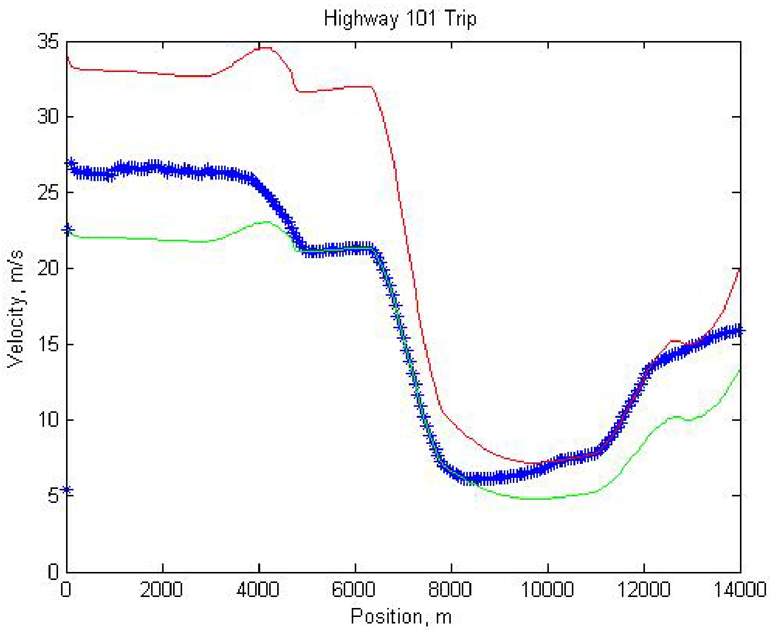
Initial Remarks

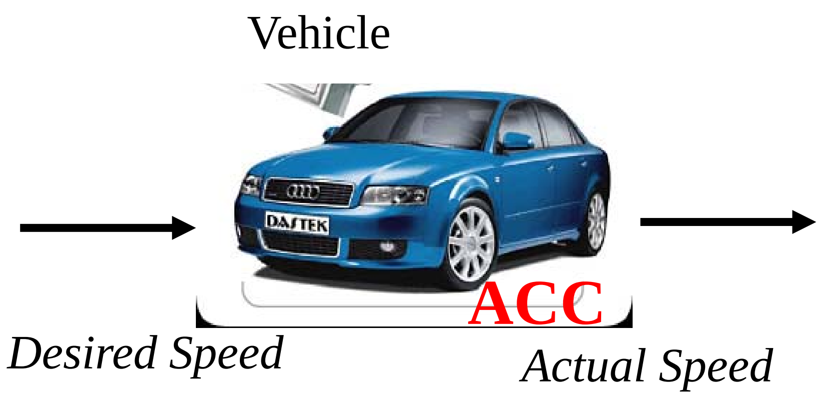
- MPC Name
- Continuous-Time versus Discrete-Time
- MPC in Practice
- **Difficulty : The theoretical side and the computation side**
- Two simple examples Models and Simulations
- Non trivial examples
- What I'd like to add: Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

Initial Remarks

- MPC Name
- Continuous-Time versus Discrete-Time
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- **Two simple examples - Models and Simulations**
- Non trivial examples
- What I’d like to add: Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

Example 1
Audi SmartEngine

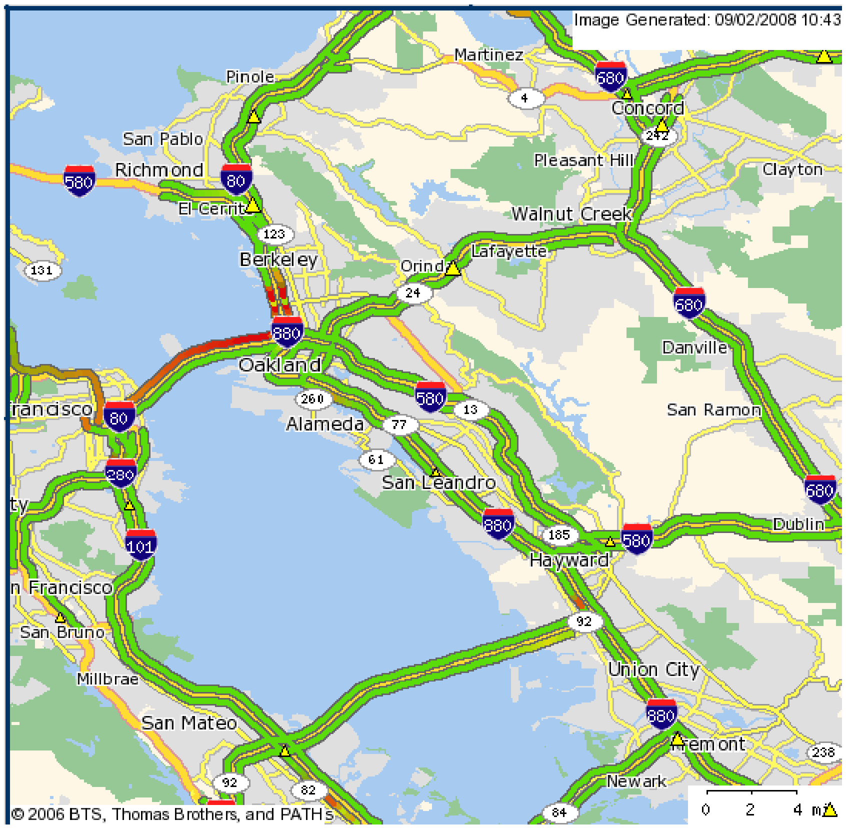




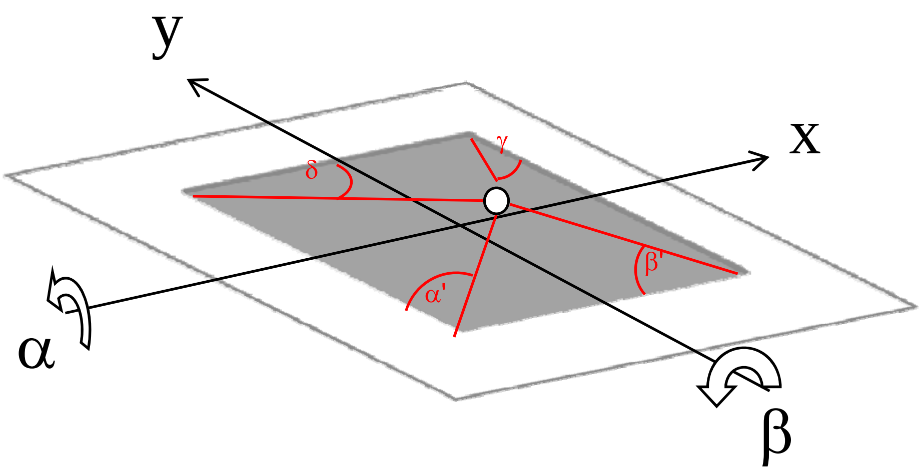
- Design and MPC Controller regulating the desired speed (through an Automatic Cruise Control) in order to reach the destination in the most fuel-efficient way
- Prediction: Max and Min Speed of traffic, Grade
- Constraints: Max and Min Speed (of traffic and of vehicle)

Example 1
Data from PeMS

- California Freeway Performance Measurement System
- Collects real-time data on CA freeways via loop detectors
- Able to communicate average traffic speed at loop location every 5 minutes
- Loops typically positioned every 0.3-3 miles



Example 2
Ball and Plate Experiment



- Specification of Experiment:

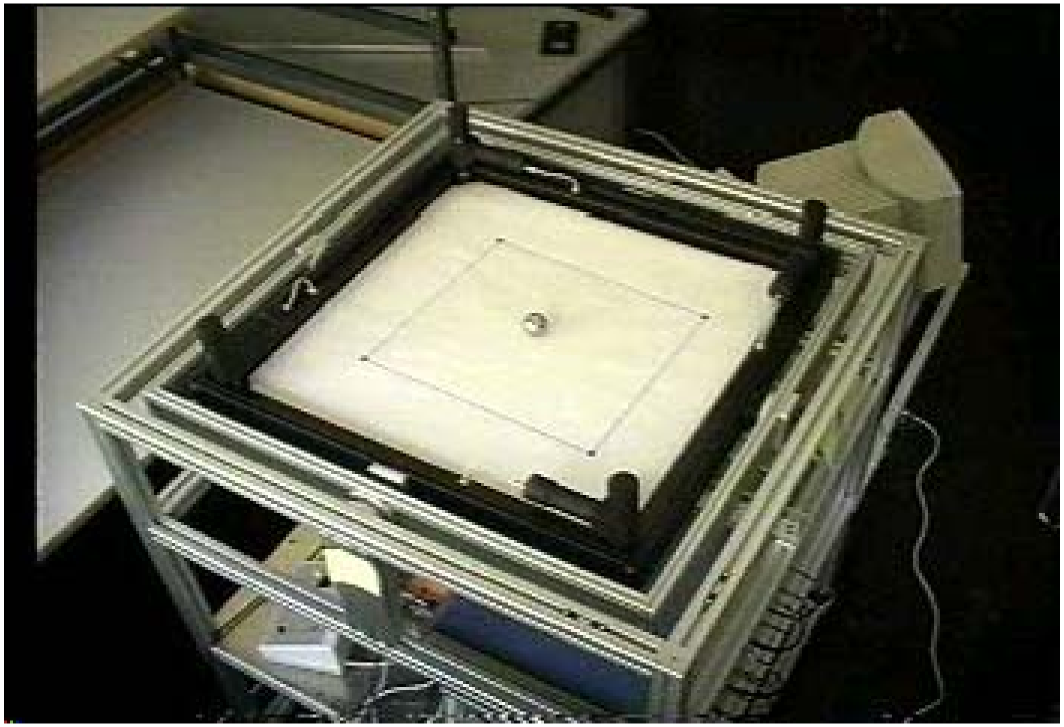
  Angle: -17°… +17° , Plate:-30 cm…+30 cm

  Input Voltage: -10 V… +10 V

  Computer: PENTIUM166

  Sampling Time: 30 ms

Example 2
Ball and Plate Experiment



Initial Remarks

- MPC Name
- Continuous-Time versus Discrete-Time
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- **Non trivial examples**
- What I'd like to add: Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

Catalytic Cracker

Open Folder

Predictive Control in NeuroScience

- **16:05 Open Folder**

Initial Remarks

- MPC Name
- Continuous-Time versus Discrete-Time
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- Non trivial examples
- **What I'd like to add:**

  Distributed, Robust, Probabilistic, Soft-Constraints
- Any preference for examples?

Initial Remarks

- MPC Name
- Continuous-Time versus Discrete-Time
- MPC in Practice
- Difficulty : The theoretical side and the computation side
- Two simple examples Models and Simulations
- Non trivial examples
- What I’d like to add:

  Distributed, Robust, Probabilistic, Soft-Constraints
- **Any preference for examples?**