Predictive Control for Linear and Hybrid Systems

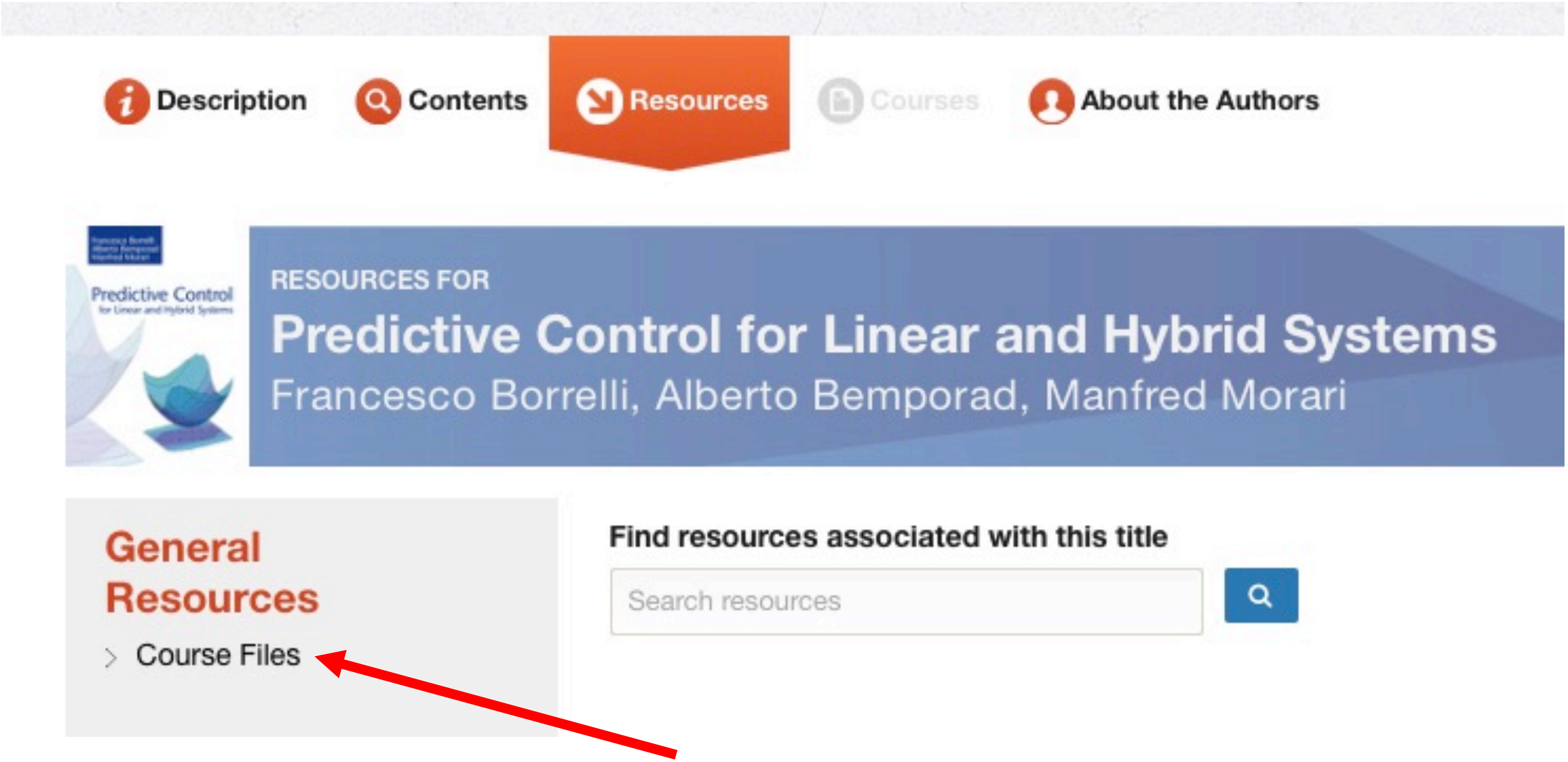
Manfred Morari
Dept. of Electrical & Systems Engineering
University of Pennsylvania

Short Course at Seoul National University
Wook Hyun Kwon Lecture
August 31, 2019

Source

Cambridge University Press

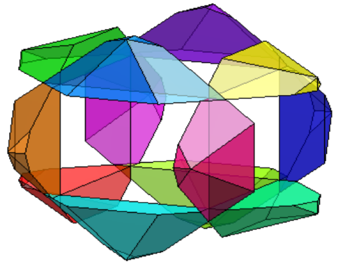
http://tinyurl.com/y2eleka8

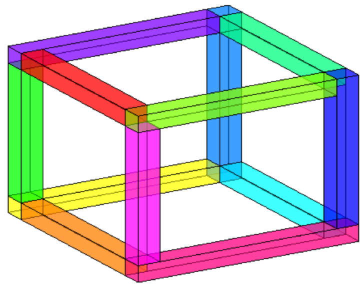


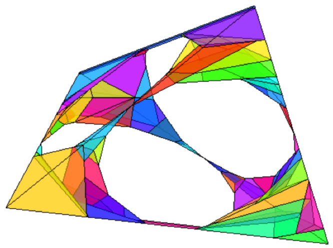
Examples & Software

- Matlab Multi-Parametric Toolbox 3 https://www.mpt3.org

- Parametric optimization
- Computational geometry features
- MPC synthesis (regulation, tracking)
  - Modeling of dynamical systems
  - Closed-loop simulations
  - Additional constraints (move blocking, soft & rate constraints, terminal sets, etc.)
  - Fine-tuning MPC setups via YALMIP
  - Code generation
  - Low-complexity explicit MPC algorithms
- Computation of invariant sets
- Construction of Lyapunov functions
- …..







Slide Co-authors

Francesco Borrelli
UC Berkeley

Colin Jones
EPF Lausanne

Melanie Zeilinger
ETH Zurich

Table of Contents

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 1: Introduction and Overview

Prof. Manfred Morari

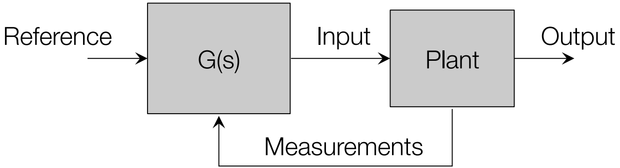
Spring 2019

Coauthors: Prof. Melanie Zeilinger, ETH Zurich
Prof. Colin Jones, EPFL
Prof. Francesco Borrelli, UC Berkeley

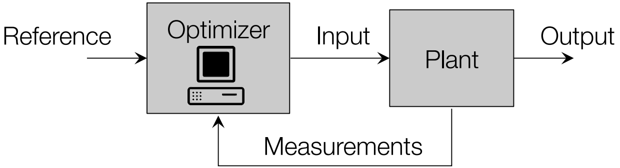
F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017.

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.

Optimization-based control: Motivation

Objective:

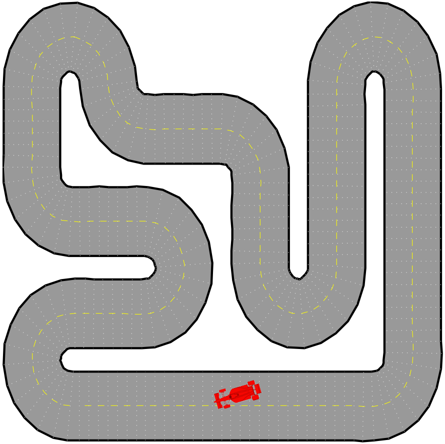
- Minimize lap time

Constraints:

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

Intuitive approach:

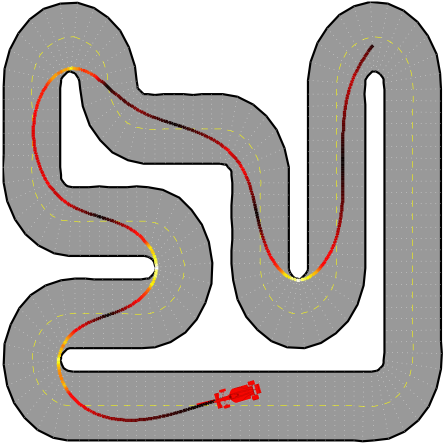
- Look forward and plan path based on
  - Road conditions
  - Upcoming corners
  - Abilities of car
  - etc...



Optimization-Based Control: Motivation

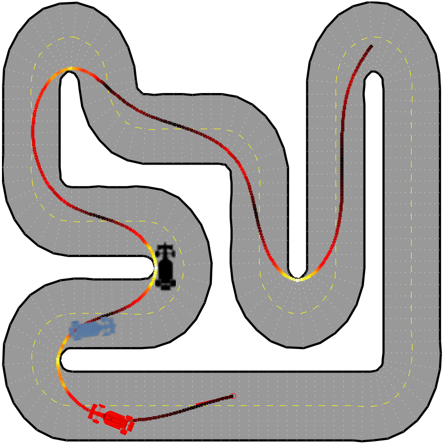
Minimize	(lap time)
while	avoid other cars
	stay on road
	...

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



Optimization-Based Control: Motivation

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

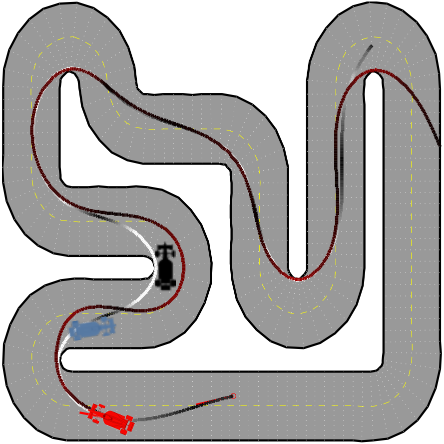


- Solve **optimization problem** to compute minimum-time path
- What to do if something unexpected happens?
  - We didn't see a car around the corner!
  - Must introduce **feedback**

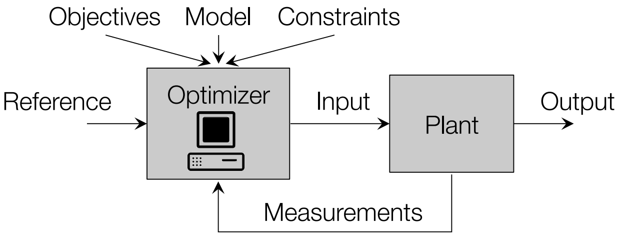
Optimization-Based Control: Motivation

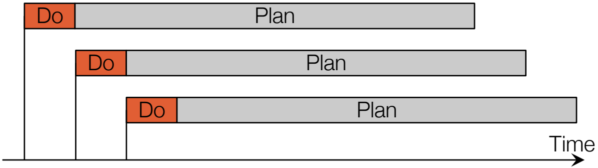
Minimize	(lap time)
while	avoid other cars
	stay on road
	...

- 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

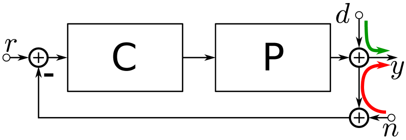




Receding horizon strategy introduces **feedback**.

Two Different Perspectives

Classical design: design C

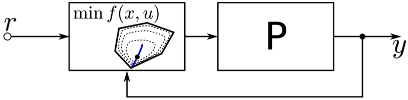


Dominant issues addressed

- 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

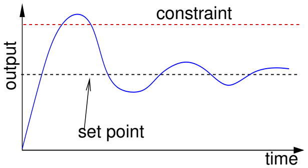
All physical systems have **constraints**:

- 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.

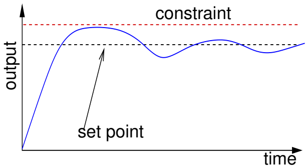
Classical control methods:

- 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



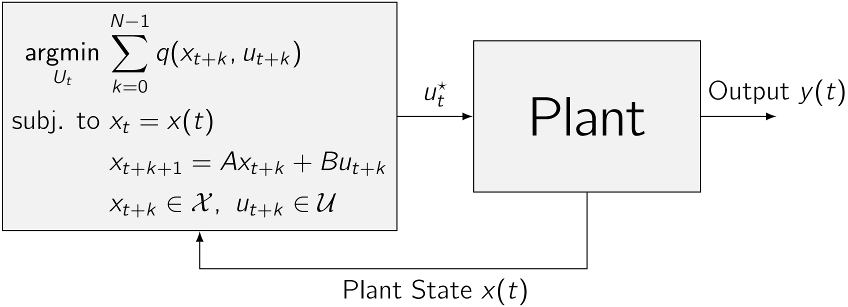
MPC: Mathematical Formulation

$$
\begin{aligned}
U_t^\star(x(t)) := \operatorname*{argmin}_{U_t} \; & \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) & \\
\text{subj. to } \; & x_t = x(t) & \text{measurement} \\
& x_{t+k+1} = Ax_{t+k} + Bu_{t+k} & \text{system model} \\
& x_{t+k} \in \mathcal{X} & \text{state constraints} \\
& u_{t+k} \in \mathcal{U} & \text{input constraints} \\
& U_t = \{u_t, u_{t+1}, \ldots, u_{t+N-1}\} & \text{optimization variables}
\end{aligned}
$$

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: $U_t^\star = \{u_t^\star, u_{t+1}^\star, \dots, u_{t+N-1}^\star\}$
- Implement only the **first** control action $u_t^\star$

Predictive Control in NeuroScience

YouTube: Charlie Rose Brain Series: The Acting Brain

Important Aspects of Model Predictive Control

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

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.
  - known as **IDCOM (Identification and Command)**
  - impulse response model for the plant, linear in inputs or internal variables (**only stable plants**)
  - quadratic performance objective over a finite prediction horizon
  - future plant output behavior specified by a reference trajectory
  - **ad hoc** input and output constraints
  - optimal inputs computed using a heuristic iterative algorithm, interpreted as the dual of identification
  - controller was not a transfer function, hence called **heuristic**

History of MPC

- 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.
  - successful in the petro-chemical industry
  - linear step response model for the plant
  - quadratic performance objective over a finite prediction horizon
  - future plant output behavior specified by trying to follow the set-point as closely as possible
  - input and output constraints included in the formulation
  - optimal inputs computed as the solution to a least–squares problem
  - **ad hoc** input and output constraints. Additional equation added online to account for constraints. Hence a **dynamic matrix** in the least squares problem.
- **C. Cutler, A. Morshedi, J. Haydel, 1983**. "An industrial perspective on advanced control". **AICHE Annual Meeting**, Washington, DC.
  - Standard QP problem formulated in order to systematically account for constraints.

History of MPC

- 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

Literature

Model Predictive Control:

- Predictive Control for linear and hybrid systems, F. Borrelli, A. Bemporad, M. Morari, 2017 Cambridge University Press
- Model Predictive Control: Theory and Design, James B. Rawlings, David Q. Mayne and Moritz M. Diehl, 2017 Nob Hill Publishing
- Receding Horizon Control, W. H. Kwon and S. Han, 2005 Springer
- Predictive Control with Constraints, Jan Maciejowski, 2000 Prentice Hall

Optimization:

- Convex Optimization, Stephen Boyd and Lieven Vandenberghe, 2004 Cambridge University Press
- Numerical Optimization, Jorge Nocedal and Stephen Wright, 2006 Springer

---

Parts of the slides in this lecture are based on or have been extracted from:

- Linear Dynamical Systems, Stephen Boyd, Stanford
- Convex Optimization, Stephen Boyd, Stanford
- Model Predictive Control, Manfred Morari, ETH Zurich
- Model Predictive Control, Colin Jones, EPFL
- Model Predictive Control, Francesco Borrelli, Berkeley

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 5: Optimal Control Introduction and Unconstrained Linear Quadratic Control

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Colin Jones, EPFL
Prof. Francesco Borrelli, UC Berkeley

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 7-9].

Outline

1. Introduction

2. Finite Horizon

3. Receding Horizon

4. Infinite Horizon

General Problem Formulation (1/2)

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.

General Problem Formulation (2/2)

Consider the **C**onstrained **F**inite **T**ime **O**ptimal **C**ontrol (CFTOC) problem.

$$
\begin{aligned}
J^\star_{0\to N}(x(0)) := \min_{U_{0\to N-1}} \ & J_{0\to N}(x(0), U_{0\to N-1}) \\
\text{subj. to } & x_{k+1} = g(x_k, u_k),\ k = 0, \ldots, N-1 \\
& h(x_k, u_k) \leq 0,\ k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(0)
\end{aligned}
$$

- $\mathcal{X}_f \subset \mathbb{R}^n$ is a *terminal* region.
- $\mathcal{X}_{0\to N} \subset \mathbb{R}^n$ is the set of feasible initial conditions $x(0)$.
- The optimal cost $J^\star_{0\to N}(x_0)$ is called *value function*.
- Assume that there exists a minimum.
- denote by $U^\star_{0\to N}$ one of the minima.

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.

Outline

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

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

  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 $U_{0\to N-1} := [u_0^\top, \ldots, u_{N-1}^\top]^\top$ that minimizes the objective function

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

- $P \succeq 0$, 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 $x_0, \ldots, x_N$ and $u_0, \ldots, u_{N-1}$ are **optimization variables** that are constrained to obey the system dynamics and the initial condition.

Batch Approach

Final Result

- The problem is unconstrained.
- Setting the gradient to zero:

$$U_0^\star(x(0)) = \mathbf{K}x(0)$$

- which implies

$$u^\star(0)(x(0)) = K_0x(0), \ldots, u^\star(N-1)(x(0)) = K_{N-1}x(0)$$

  which is a linear, open-loop controller function of the initial state $x(0)$.
- The optimal cost is

$$J_0^\star(x(0)) = x^\top(0)P_0x(0)$$

  which is a positive definite quadratic function of the initial state $x(0)$.

Recursive Approach

Final Result

- The problem is unconstrained
- Using the Dynamic Programming Algorithm we have,

$$u^\star(k) = F_k x(k)$$

  which is a linear, time-varying state-feedback controller.
- the optimal cost-to-go $k \to N$ is

$$J_k^\star(x(k)) = x^\top(k) P_k x(k)$$

  which is a positive definite quadratic function of the state at time k.
- F_k is computed by using P_{k+1}
- Each P_k is related to P_{k+1} by a recursive equation (Riccati Difference Equation)

Comparison of Batch and Recursive Approaches (1/2)

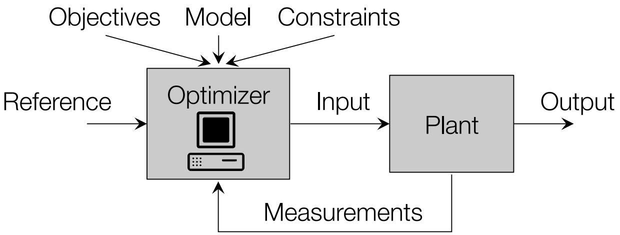
- Fundamental difference: Batch optimization returns a sequence $U^\star(x(0))$ of **numeric values** depending only on the initial state $x(0)$, while dynamic programming yields **feedback policies** $u_k^\star = F_k x_k$, $k = 0, \ldots, N-1$ depending on each x_k.
- If the state evolves exactly as modelled, then the sequences of control actions obtained from the two approaches are identical.
- The recursive solution should be more robust to disturbances and model errors, because if the future states later deviate from their predicted values, the exact optimal input can still be computed.
- The Recursive Approach is computationally more attractive because it breaks the problem down into single-step problems. For large horizon length, the Hessian H in the Batch Approach, which must be inverted, becomes very large.

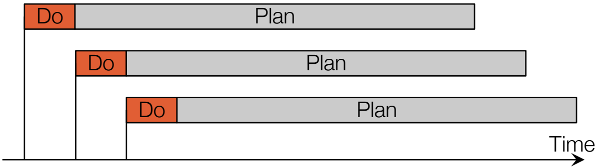
Comparison of Batch and Recursive Approaches (2/2)

- Without any modification, both solution methods will break down when inequality constraints on x_k or u_k are added.
- The Batch Approach is far easier to adapt than the Recursive Approach when constraints are present: just perform a constrained minimization for the current state.
- Doing this at **every** time step within the time available, and then using only the first input from the resulting sequence, amounts to **receding horizon control**.

Outline

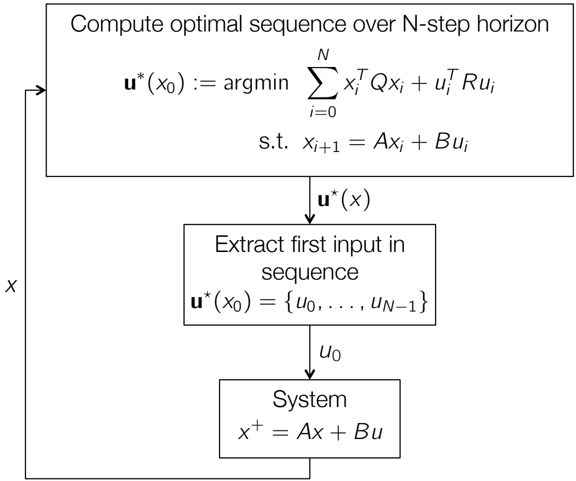
Receding horizon control





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

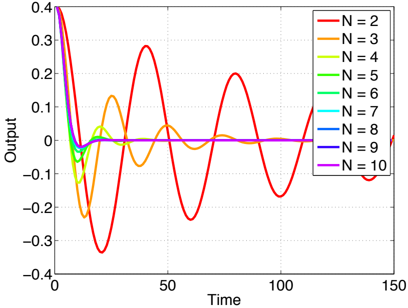
$$G(s) := \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

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

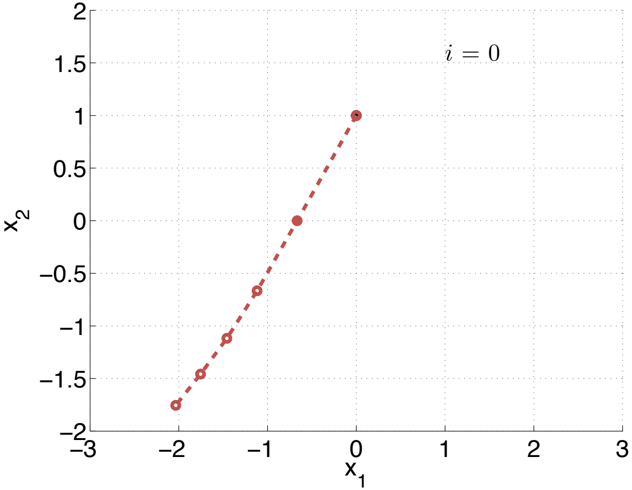
Discrete-time state-space model:

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

Closed-loop response

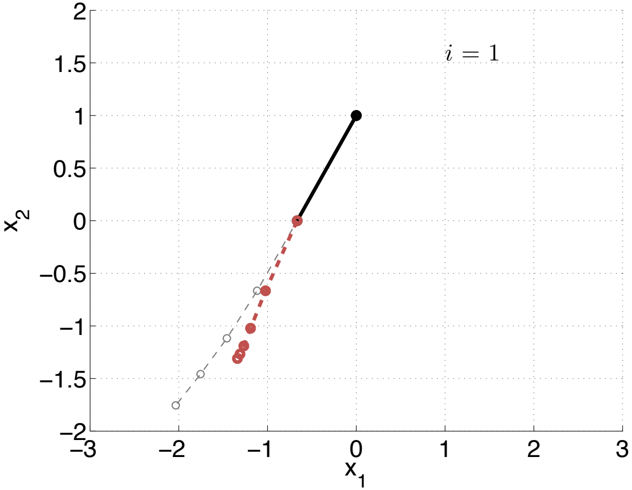


Example: Short horizon $N = 5$



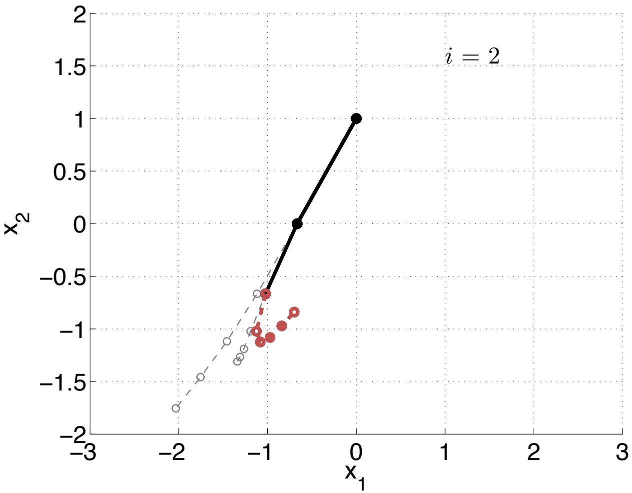
Short horizon: Prediction and closed-loop response differ.

Example: Short horizon $N = 5$



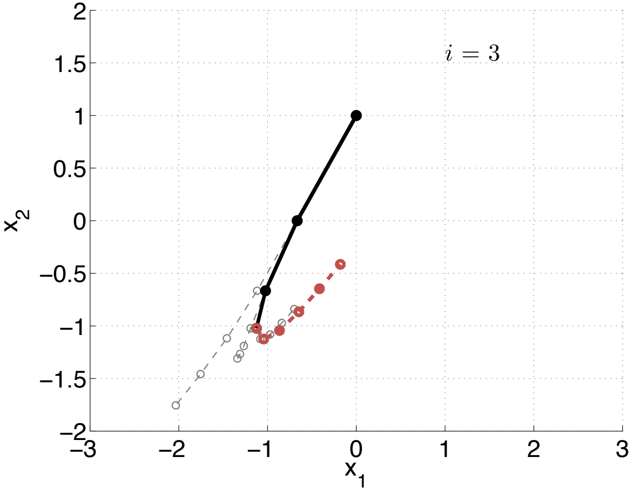
Short horizon: Prediction and closed-loop response differ.

Example: Short horizon $N = 5$



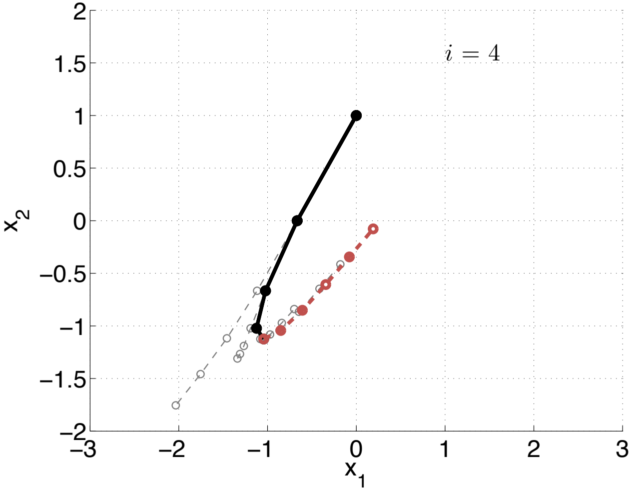
Short horizon: Prediction and closed-loop response differ.

Example: Short horizon $N = 5$



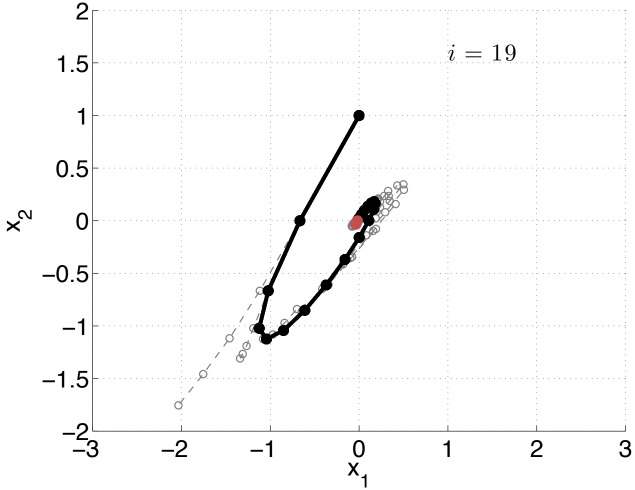
Short horizon: Prediction and closed-loop response differ.

Example: Short horizon $N = 5$



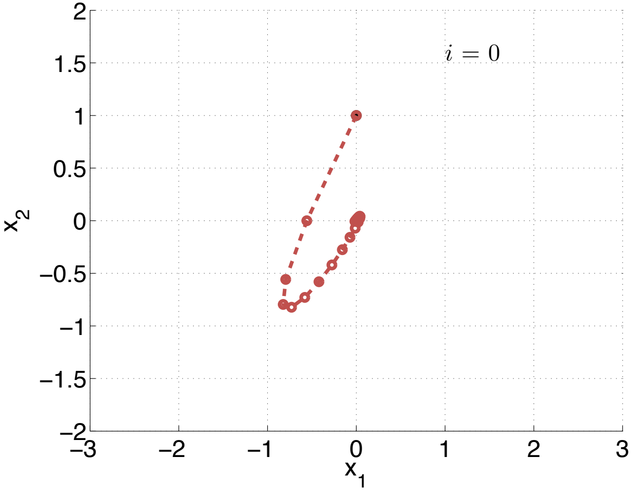
Short horizon: Prediction and closed-loop response differ.

Example: Short horizon $N = 5$



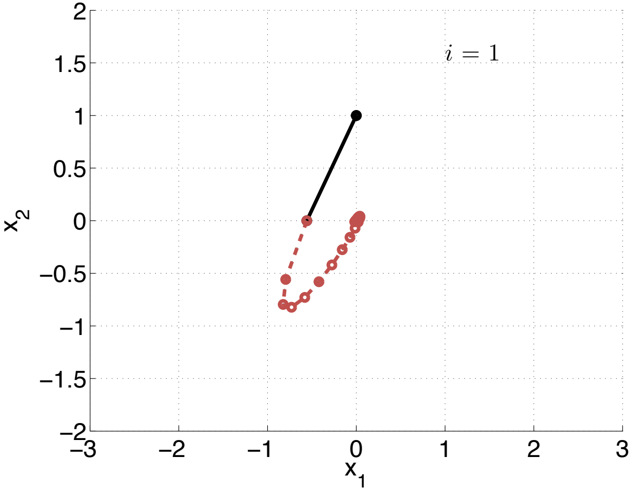
Short horizon: Prediction and closed-loop response differ.

Example: Long horizon $N = 20$



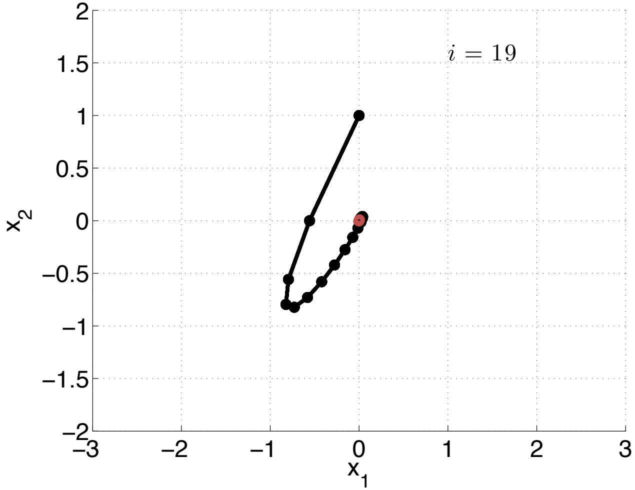
Long horizon: Prediction and closed-loop match.

Example: Long horizon $N = 20$



Long horizon: Prediction and closed-loop match.

Example: Long horizon $N = 20$



Long horizon: Prediction and closed-loop match.

Stability of Finite-Horizon Optimal Control Laws

Consider the system

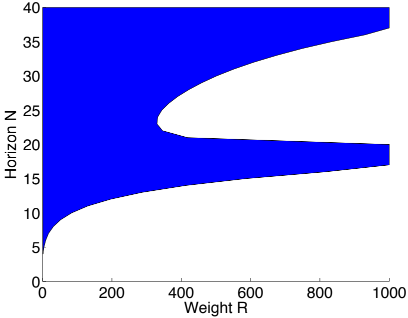
$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

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



Outline

Infinite Horizon Control Problem: Optimal Solution (1/2)

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

$$
\begin{aligned}
J_\infty(x(0)) = \min_{u(\cdot)} & \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) \\
\text{subj. to } & x_{k+1} = A x_k + B u_k, \quad k = 0, 1, 2, \ldots, \infty\,, \\
& x_0 = x(0)
\end{aligned}
$$

- 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 (2/2)

- 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.

Stability of Infinite-Horizon LQR

- In addition, the closed-loop system with $u(k) = F_\infty x(k)$ is guaranteed to be asymptotically stable, under the stabilizability and detectability assumptions of the previous slide.
- The latter statement can be proven by substituting the control law $u(k) = F_\infty x(k)$ into $x(k+1) = Ax(k) + Bu(k)$, and then examining the properties of the system

$$x(k+1) = (A + BF_\infty)x(k)\,. \tag{7}$$

- The asymptotic stability of (7) can be proven by showing that the infinite horizon cost $J^\star_\infty(x(k)) = x(k)^\top P_\infty x(k)$ is actually a Lyapunov function for the system, i.e. $J^\star_\infty(x(k)) > 0$, $\forall k \neq 0$, $J^\star_\infty(0) = 0$, and $J^\star_\infty(x(k+1)) < J^\star_\infty(x(k))$, for any $x(k)$. This implies that

$$\lim_{k\to\infty} x(k) = 0\,.$$

Choices of Terminal Weight P in Finite Horizon Control (1/2)

1. The terminal cost P of the finite horizon problem can in fact trivially be chosen so that its solution matches the infinite horizon solution
   - To do this, make P equal to the optimal cost from N to ∞ (i.e. the cost with the optimal controller choice). This can be computed from the ARE:

$$P = A^\top PA + Q - A^\top PB(B^\top PB + R)^{-1}B^\top PA$$

   - This approach rests on the assumption that no constraints will be active after the end of the horizon.

Choices of Terminal Weight P in Finite Horizon Control (2/2)

2. Choose P assuming no control action after the end of the horizon, so that

$$x(k+1) = Ax(k),\ k = N, \ldots, \infty$$

    - This P can be determined from solving the Lyapunov equation

$$APA^\top + Q = P\,.$$

    - This approach only makes sense if the system is asymptotically stable (or no positive definite solution P will exist).

3. Assume we want the state and input both to be zero after the end of the finite horizon. In this case no P but an extra constraint is needed

$$x_N = 0$$

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 6: Constrained Finite Time Optimal Control

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Colin Jones, EPFL
Prof. Melanie Zeilinger, ETH Zurich
Prof. Francesco Borrelli, UC Berkeley

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 11].

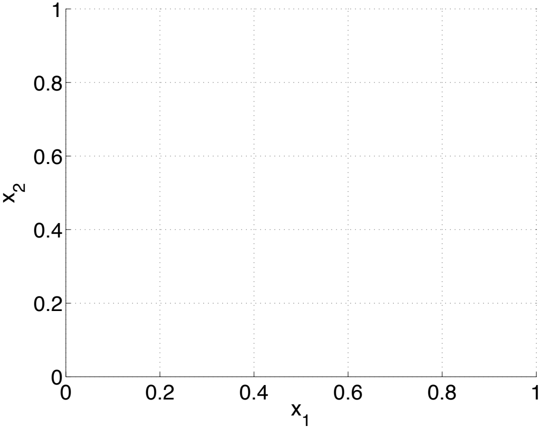
Objectives of Constrained Optimal Control

$$x^+ = f(x,u) \qquad\qquad (x,u) \in \mathcal{X}, \mathcal{U}$$

Design control law $u = \kappa(x)$ such that the system:

1. Satifies constraints : $\{x_i\} \subset \mathcal{X}$, $\{u_i\} \subset \mathcal{U}$
2. Is asymptotically stable: $\lim_{i\to\infty} x_i = 0$
3. Optimizes "performance"
4. Maximizes the set $\{x_0 \mid \text{Conditions 1-3 are met}\}$

Limitations of Linear Controllers



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

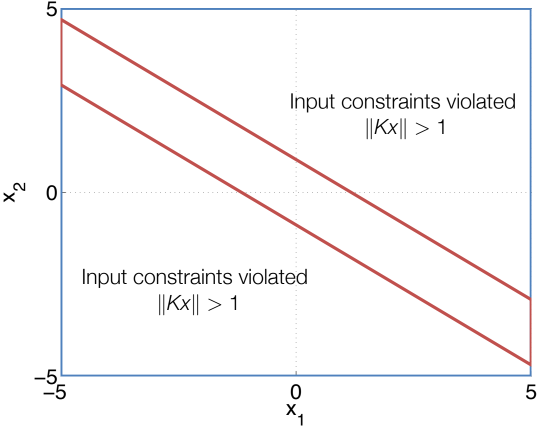
Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \leq 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \leq 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Limitations of Linear Controllers



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

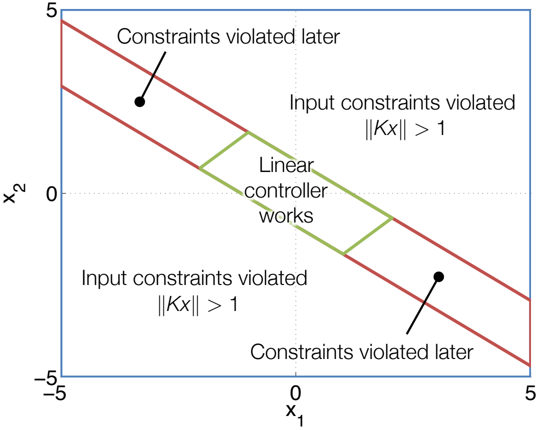
Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \leq 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \leq 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Limitations of Linear Controllers



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

Constraints:

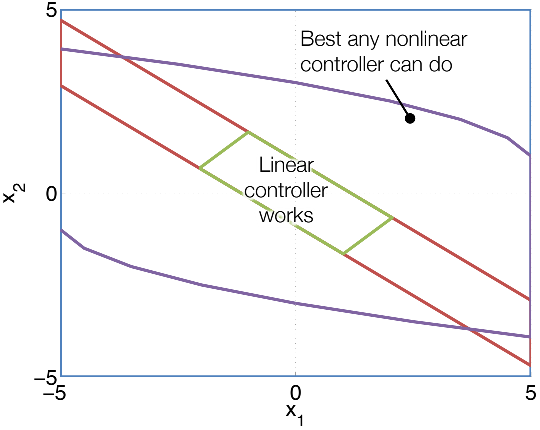
$$\mathcal{X} := \{x \mid \|x\|_\infty \le 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Yes, but the region where it works is very small

Limitations of Linear Controllers



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \le 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Yes, but the region where it works is very small

Use nonlinear control (MPC) to increase the region of attraction

Constrained Infinite Time Optimal Control (what we would like to solve)

$$
\begin{aligned}
J_0^*(x(0)) = \min \ & \sum_{k=0}^{\infty} q(x_k, u_k) \\
\text{s.t. } & x_{k+1} = Ax_k + Bu_k, k = 0, \dots, N-1 \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, k = 0, \dots, N-1 \\
& x_0 = x(0)
\end{aligned}
$$

- **Stage cost** $q(x, u)$: "cost" of being in state x and applying input u
- Optimizing over a trajectory provides a **tradeoff between short- and long-term benefits** of actions
- We'll see that such a control law has many beneficial properties...
  ... but we can't compute it: there are an **infinite number of variables**

Constrained Finite Time Optimal Control (what we can sometimes solve)

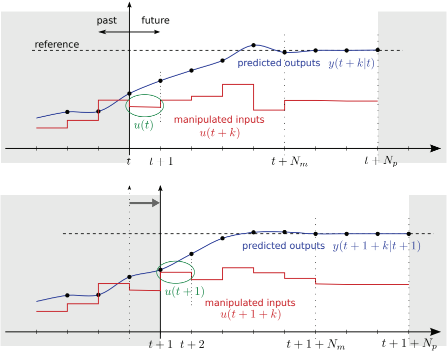
$$
\begin{aligned}
J_t^*(x(t)) = \quad \min_{U_t} \quad & p(x_{t+N}) + \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) \\
\text{subj. to} \quad & x_{t+k+1} = Ax_{t+k} + Bu_{t+k},\ \ k = 0, \ldots, N-1 \\
& x_{t+k} \in \mathcal{X},\ \ u_{t+k} \in \mathcal{U},\ \ k = 0, \ldots, N-1 \\
& x_{t+N} \in \mathcal{X}_f \\
& x_t = x(t)
\end{aligned} \qquad (1)
$$

where $\mathcal{U}_t = \{u_t, \ldots, u_{t+N-1}\}$.

Truncate after a finite horizon:

- $p(x_{t+N})$: Approximates the 'tail' of the cost
- $\mathcal{X}_f$: Approximates the 'tail' of the constraints

On-line Receding Horizon Control



1. At each sampling time, solve a **CFTOC**.
2. Apply the optimal input **only during** $[t, t+1]$
3. At $t+1$ solve a CFTOC over a **shifted horizon** based on new state measurements
4. The resulting controller is referred to as **Receding Horizon Controller (RHC)** or **Model Predictive Controller (MPC)**.

On-line Receding Horizon Control

1) MEASURE the state $x(t)$ at time instance t
2) OBTAIN $U_t^*(x(t))$ by solving the optimization problem in (1)
3) IF $U_t^*(x(t)) = \emptyset$ THEN 'problem infeasible' STOP
4) APPLY the first element u_t^* of U_t^* to the system
5) WAIT for the new sampling time $t+1$, GOTO 1)

Note that we need a constrained optimization solver for step 2).

MPC Features

Pros

- Any model:
  - linear
  - nonlinear
  - single/multivariable
  - time delays
  - constraints
- Any objective:
  - sum of squared errors
  - sum of absolute errors (i.e., integral)
  - worst error over time
  - economic objective

Cons

- Computationally demanding in the general case
- May or may not be stable
- May or may not be feasible

Problem Formulation

Quadratic cost function

$$J_0(x(0), U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \tag{3}$$

with $P \succeq 0$, $Q \succeq 0$, $R \succ 0$.
Constrained Finite Time Optimal Control problem (CFTOC).

$$\begin{array}{lll} J_0^*(x(0)) = & \min\limits_{U_0} & J_0(x(0), U_0) \\ & \text{subj. to} & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_0 = x(0) \end{array} \tag{4}$$

N is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

Construction of the QP with substitution

- **Step 1**: Rewrite the cost as

$$\begin{aligned} J_0(x(0), U_0) &= U_0' H U_0 + 2x(0)' F U_0 + x(0)' Y x(0) \\ &= [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0' \; x(0)']' \end{aligned}$$

  Note: $\left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] \succeq 0$ since $J_0(x(0), U_0) \geq 0$ by assumption.

- **Step 2**: Rewrite the constraints compactly as (details provided on the next slide)

$$G_0 U_0 \leq w_0 + E_0 x(0)$$

- **Step 3**: Rewrite the optimal control problem as

$$\begin{aligned} J_0^*(x(0)) = \min_{U_0} \quad & [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0' \; x(0)']' \\ \text{subj. to} \quad & G_0 U_0 \leq w_0 + E_0 x(0) \end{aligned}$$

Solution

$$J_0^*(x(0)) = \min_{U_0} \quad [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']'$$
$$\text{subj. to} \quad G_0 U_0 \leq w_0 + E_0 x(0)$$

For a given x(0) U_0^* can be found via a QP solver.

2-Norm State Feedback Solution

Start from QP with substitution.

- **Step 1**: Define $z \triangleq U_0 + H^{-1}F'x(0)$ and transform the problem into

$$\begin{array}{lll} \hat{J}^*(x(0)) = & \min\limits_{z} & z'Hz \\ & \text{subj. to} & G_0 z \leq w_0 + S_0 x(0), \end{array}$$

  where $S_0 \triangleq E_0 + G_0 H^{-1} F'$, and
  $\hat{J}^*(x(0)) = J_0^*(x(0)) - x(0)'(Y - FH^{-1}F')x(0)$.

  The CFTOC problem is now a **multiparametric quadratic program (mp-QP)**.

- **Step 2**: Solve the mp-QP to get explicit solution $z^*(x(0))$

- **Step 3**: Obtain $U_0^*(x(0))$ from $z^*(x(0))$

2-Norm State Feedback Solution

Main Results

1. The **open loop optimal control function** can be obtained by solving the mp-QP problem and calculating $U_0^*(x(0)),\ \forall x(0) \in \mathcal{X}_0$ as $U_0{}^* = z^*(x(0)) - H^{-1}F'x(0)$.
2. The first component of the multiparametric solution has the form
$$u^*(0) = f_0(x(0)),\ \ \forall x(0) \in \mathcal{X}_0,$$
$f_0 : \mathbb{R}^n \rightarrow \mathbb{R}^m$, is continuous and PieceWise Affine on Polyhedra
$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \dots, N_0^r$$
3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \dots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. The value function $J_0^*(x(0))$ is convex and piecewise quadratic on polyhedra.

Example

Consider the double integrator

$$\begin{cases} x(t+1) & = & \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) & = & \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

subject to constraints

$$-1 \le u(k) \le 1,\ k = 0, \dots, 5$$

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \le x(k) \le \begin{bmatrix} 10 \\ 10 \end{bmatrix},\ k = 0, \dots, 5$$

Compute the **state feedback** optimal controller $u^*(0)(x(0))$ solving the

CFTOC problem with $N = 6$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.1$, P the solution of the ARE, $\mathcal{X}_f = \mathbb{R}^2$.

Example

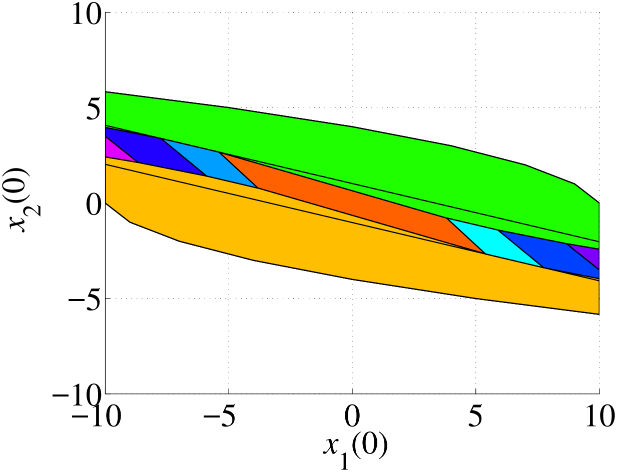


Figure: Partition of the state space for the affine control law $u^*(0)$ ($N_0^r = 13$)

Example

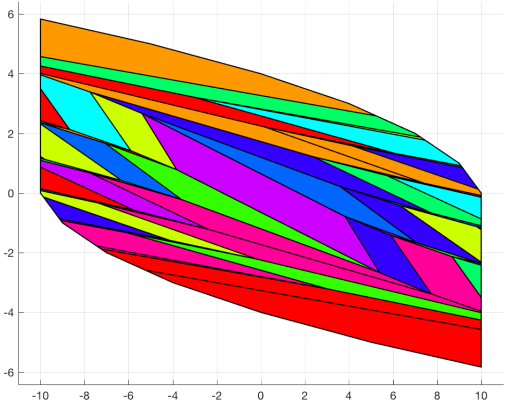


Figure: Partition of the state space for the affine control law $u^*(0)$ ($N_0^r = 61$)

Example

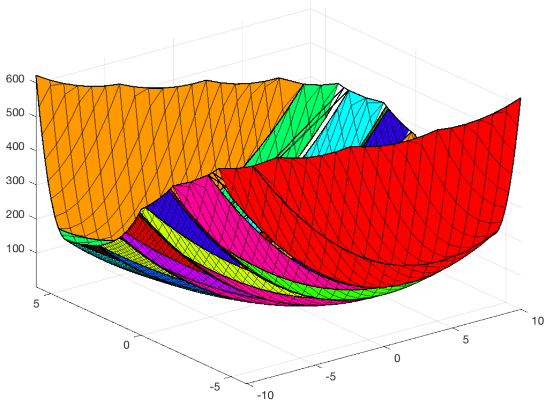


Figure: Value function for the affine control law $u^*(0)$ ($N_0^r = 61$)

Example

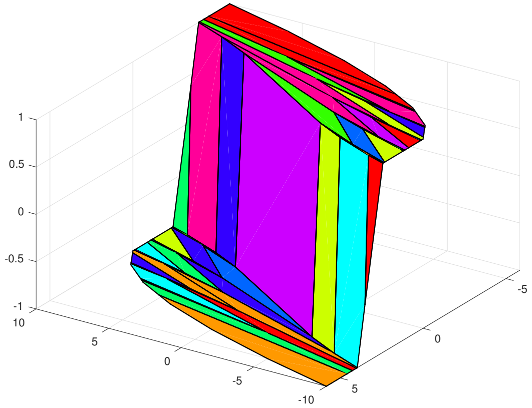


Figure: Optimal control input for the affine control law $u^*(0)$ ($N_0^r = 61$)

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 7: Guaranteeing Feasibility and Stability

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Colin Jones, EPFL
Prof. Melanie Zeilinger, ETH Zurich
Prof. Francesco Borrelli, UC Berkeley

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 12].

Infinite Time Constrained Optimal Control (what we would like to solve)

$$
\begin{aligned}
J_0^\star(x(0)) = \min \ & \sum_{k=0}^{\infty} q(x_k, u_k) \\
\text{subj. to } & x_{k+1} = Ax_k + Bu_k, k = 0, 1, 2, \ldots \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, k = 0, 1, 2, \ldots \\
& x_0 = x(0)
\end{aligned}
$$

- **Stage cost** $q(x, u)$ describes "cost" of being in state x and applying input u.
- Optimizing over a trajectory provides a **tradeoff between short- and long-term benefits** of actions
- We'll see that such a control law has many beneficial properties...
  ...but we can't compute it: there are an **infinite number of variables**

Receding Horizon Control (what we can sometimes solve)

$$
\begin{aligned}
J_t^\star(x(t)) = \min_{U_t} \; & p(x_{t+N}) + \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) \\
\text{subj. to } & x_{t+k+1} = Ax_{t+k} + Bu_{t+k}, k = 0, \ldots, N-1 \\
& x_{t+k} \in \mathcal{X}, u_{t+k} \in \mathcal{U}, k = 0, \ldots, N-1 \\
& x_{t+N} \in \mathcal{X}_f \\
& x_t = x(t)
\end{aligned}
$$

where $\mathcal{U}_t = \{u_t, \ldots, u_{t+N-1}\}$.

Truncate after a finite horizon:

- $p(x_{t+N})$: Approximates the 'tail' of the cost
- $\mathcal{X}_f$: Approximates the 'tail' of the constraints

Example: Loss of feasibility - Double Integrator

Consider the double integrator

$$\begin{cases} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

subject to the input constraints

$$-0.5 \leq u(t) \leq 0.5$$

and the state constraints

$$\begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x(t) \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}.$$

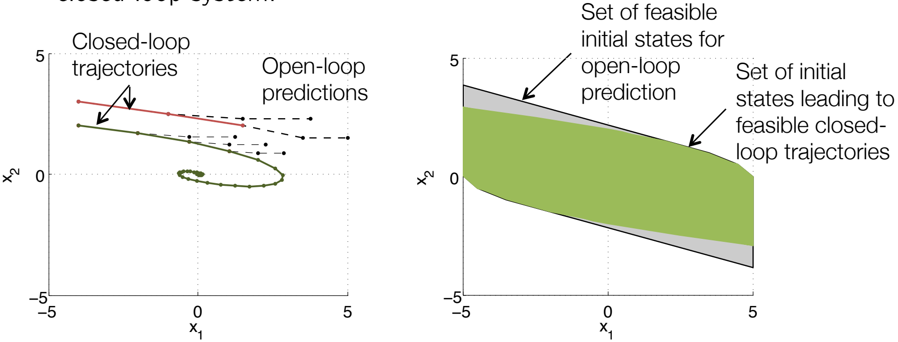
Compute a receding horizon controller with quadratic objective with

$$N = 3,\ P = Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ R = 10.$$

Summary: Feasibility and Stability

Problems originate from the use of a ‘short sighted’ strategy

⇒ Finite horizon causes deviation between the open-loop prediction and the closed-loop system:



Ideally we would solve the MPC problem with an infinite horizon, but that is computationally intractable

⇒ Design finite horizon problem such that it approximates the infinite horizon

Summary: Feasibility and Stability

- Infinite-Horizon
  If we solve the RHC problem for $N = \infty$ (as done for LQR), then the open loop trajectories are the same as the closed loop trajectories. Hence
  - If problem is feasible, the closed loop trajectories will be always feasible
  - If the cost is finite, then states and inputs will converge asymptotically to the origin

- Finite-Horizon
  RHC is “short-sighted” strategy approximating infinite horizon controller. But
  - **Feasibility**. After some steps the finite horizon optimal control problem may become infeasible. (Infeasibility occurs without disturbances and model mismatch!)
  - **Stability**. The generated control inputs may not lead to trajectories that converge to the origin.

Feasibility and stability in MPC - Solution

Main idea: Introduce terminal cost and constraints to explicitly ensure feasibility and stability:

$$
\begin{aligned}
J_0^*(x_0) = \quad & \min_{U_0} \quad p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k) && \text{Terminal Cost} \\
& \text{subj. to} \\
& \quad x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\
& \quad x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\
& \quad x_N \in \mathcal{X}_f && \text{Terminal Constraint} \\
& \quad x_0 = x(t)
\end{aligned}
$$

$p(\cdot)$ and $\mathcal{X}_f$ are chosen to mimic an infinite horizon.

Stability of MPC - Main Result

Assumptions

1. Stage cost is positive definite, i.e. it is strictly positive and only zero at the origin
2. Terminal set is **invariant** under the local control law $v(x_k)$:

$$x_{k+1} = Ax_k + Bv(x_k) \in \mathcal{X}_f, \quad \text{for all } x_k \in \mathcal{X}_f$$

   All state and input **constraints are satisfied** in $\mathcal{X}_f$:

$$\mathcal{X}_f \subseteq \mathcal{X},\ v(x_k) \in \mathcal{U}, \quad \text{for all } x_k \in \mathcal{X}_f$$

3. Terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_f$ and satisfies:

$$p(x_{k+1}) - p(x_k) \leq -q(x_k, v(x_k)), \quad \text{for all } x_k \in \mathcal{X}_f$$

Under those 3 assumptions:

Theorem

The closed-loop system under the MPC control law $u_0^*(x)$ is asymptotically stable and the set $\mathcal{X}_f$ is positive invariant for the system

$$x(k+1) = Ax + Bu_0^*(x).$$

MPC Stability and Feasibility - Summary

IF we choose: $\mathcal{X}_f$ to be an invariant set (Assumption 2) and the terminal cost $p(x)$ to be a Lyapunov function with the decrease described in Assumption 3, THEN

- The set of feasible initial states $\mathcal{X}_0$ is also the set of initial states which are persistently feasible (feasible for all $t \geq 0$) for the system in closed-loop with the designed MPC.
- The equilibrium point $(0, 0)$ is asymptotically stable according to Lyapunov.
- $J_0^\star(x)$ is a Lyapunov function for the closed loop system (system + MPC) defined over $\mathcal{X}_0$. Then $\mathcal{X}_0$ is the region of attraction of the equilibrium point.
- Proof works for any nonlinear system and positive definite and continuous stage cost as long as the optimizer is unique.

Choice of Terminal Sets and Cost - Linear System, Quadratic Cost

- Design unconstrained LQR control law

$$F_\infty = -(B' P_\infty B + R)^{-1} B' P_\infty A$$

where P_∞ is the solution to the discrete-time algebraic Riccati equation:

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

- Choose the terminal weight $P = P_\infty$
- Choose the terminal set $\mathcal{X}_f$ to be the maximum invariant set for the closed-loop system $x_{k+1} = (A + BF_\infty) x_k$:

$$x_{k+1} = Ax_k + BF_\infty(x_k) \in \mathcal{X}_f, \quad \text{for all } x_k \in \mathcal{X}_f$$

All state and input **constraints are satisfied** in $\mathcal{X}_f$:

$$\mathcal{X}_f \subseteq \mathcal{X},\ F_\infty x_k \in \mathcal{U}, \quad \text{for all } x_k \in \mathcal{X}_f$$

Choice of Terminal Sets and Cost - Linear System, Quadratic Cost

1. The stage cost is a positive definite function
2. By construction the terminal set is **invariant** under the local control law $v = F_\infty x$
3. Terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_f$ and satisfies:

$$
\begin{aligned}
& x'_{k+1} P x_{k+1} - x'_k P x_k \\
& = x'_k(-P_\infty + A' P_\infty A - A' P_\infty B (B' P_\infty B + R)^{-1} B' P_\infty A - F'_\infty R F_\infty) x_k \\
& = -x'_k Q x_k - v'_k R v_k
\end{aligned}
$$

All the Assumptions of the Feasibility and Stability Theorem are verified.

Example: Unstable Linear System

System dynamics:

$$x_{k+1} = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u_k$$

Constraints:

$$\mathcal{X} := \{x \mid -50 \le x_1 \le 50,\ -10 \le x_2 \le 10\} = \{x \mid A_x x \le b_x\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\} = \{u \mid A_u u \le b_u\}$$

Stage cost:

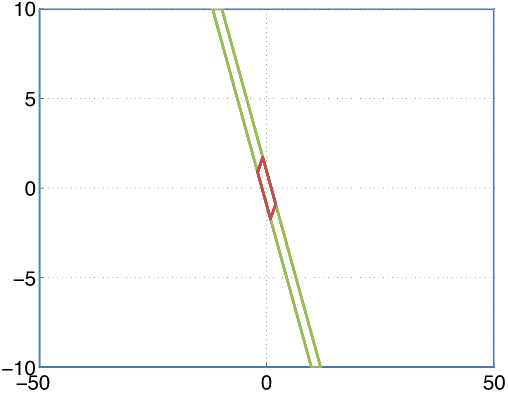
$$q(x,u) := x' \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + u^\top u$$

Horizon: $N = 10$

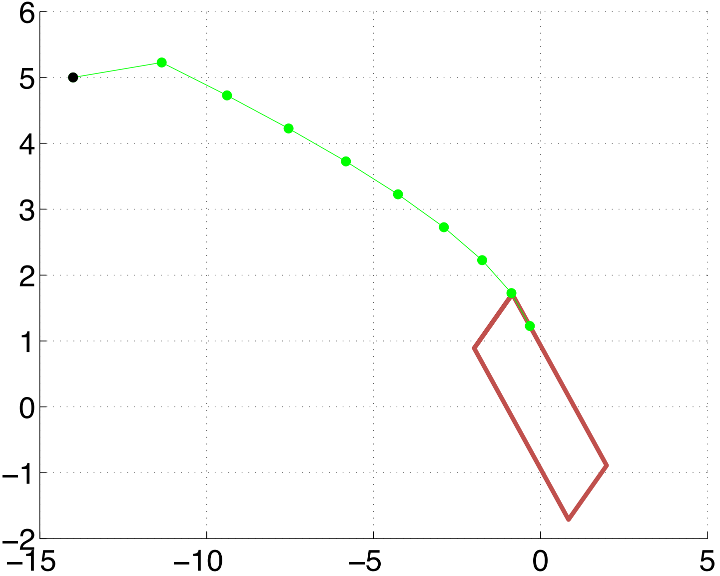
Example: Designing MPC Problem

1. Compute the optimal LQR controller and cost matrices: F_∞, P_∞
2. Compute the maximal invariant set $\mathcal{X}_f$ for the closed-loop linear system $x_{k+1} = (A + BF_\infty)x_k$ subject to the constraints

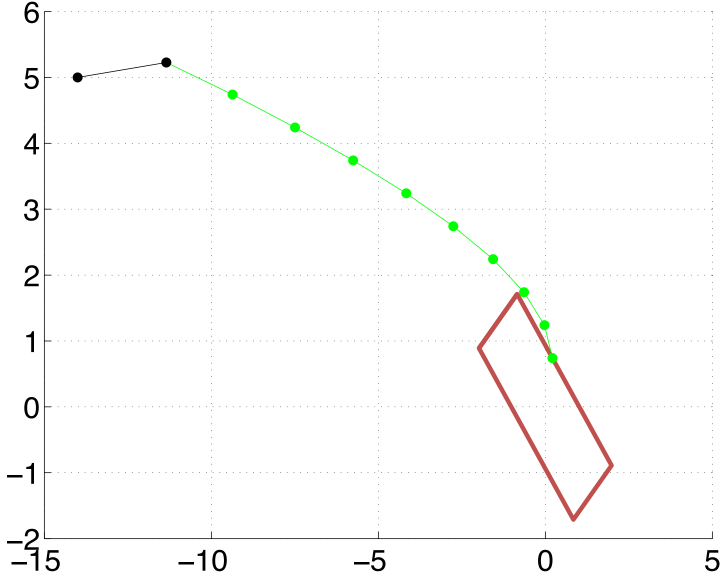
$$\mathcal{X}_{\text{cl}} := \left\{ x \;\middle|\; \begin{bmatrix} A_x \\ A_u F_\infty \end{bmatrix} x \leq \begin{bmatrix} b_x \\ b_u \end{bmatrix} \right\}$$



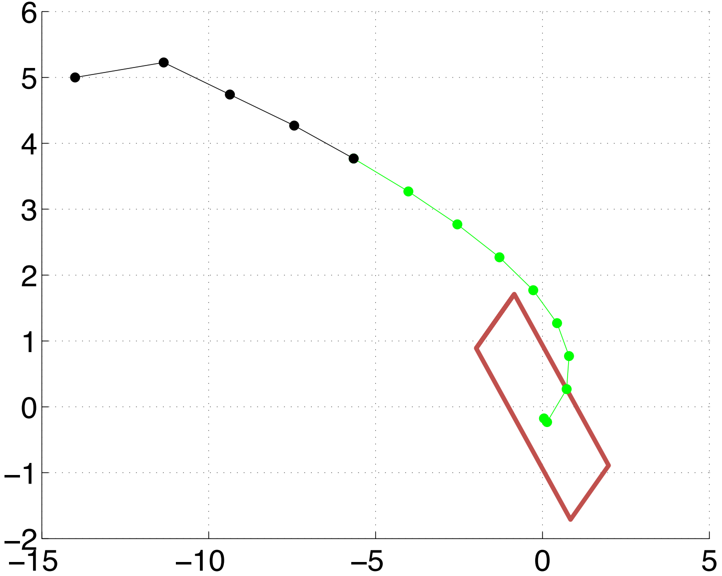
Example: Closed-loop behaviour



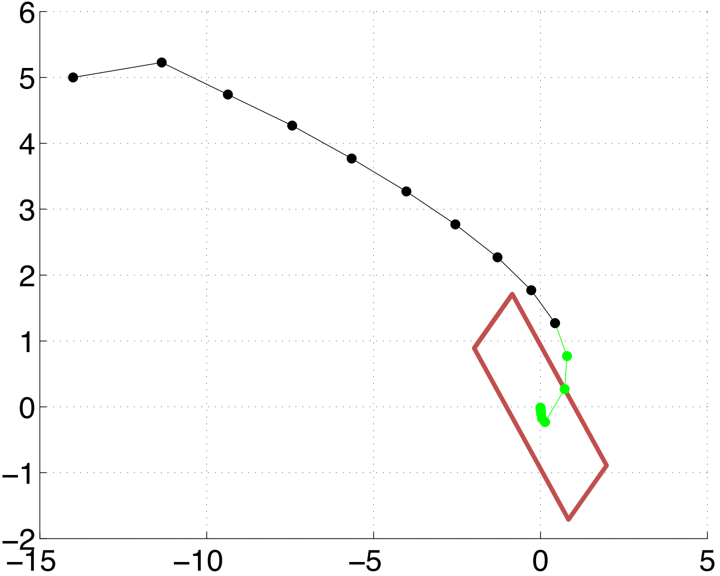
Example: Closed-loop behaviour



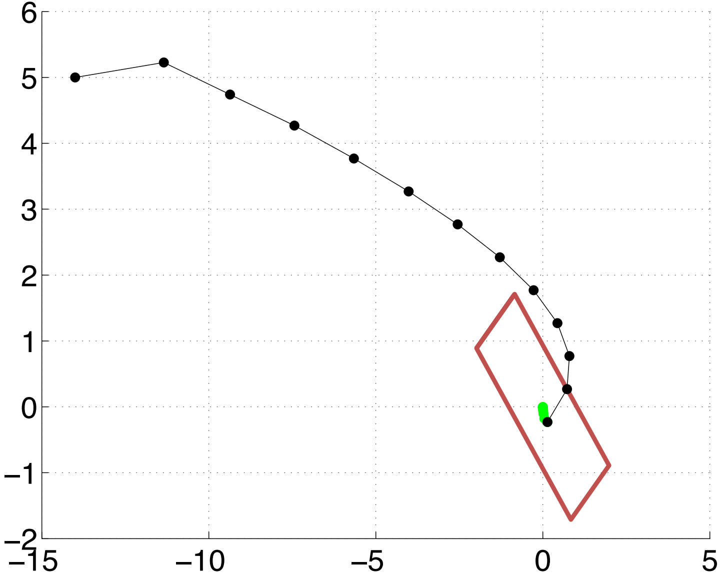
Example: Closed-loop behaviour



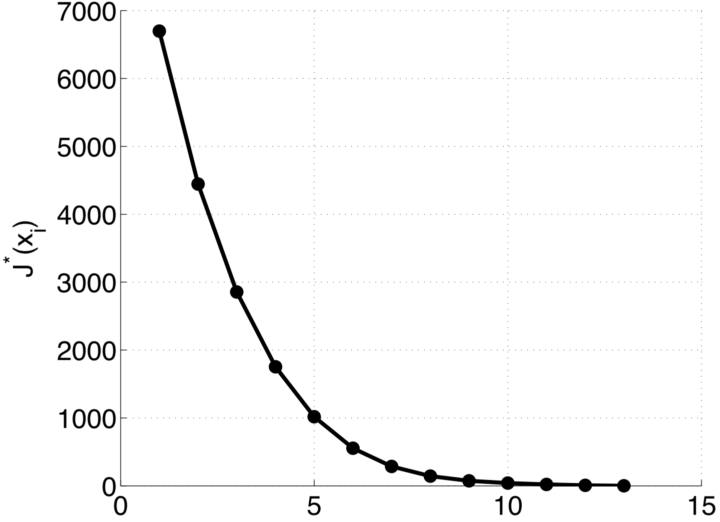
Example: Closed-loop behaviour



Example: Closed-loop behaviour



Example: Lyapunov Decrease of Optimal Cost



Stability of MPC - Remarks

- The terminal set $\mathcal{X}_f$ and the terminal cost ensure recursive feasibility and stability of the closed-loop system.
  But: the terminal constraint reduces the region of attraction.
  (Can extend the horizon to a sufficiently large value to increase the region)

Are terminal sets used in practice?

- Generally not...
  - Not well understood by practitioners
  - Requires advanced tools to compute (polyhedral computation or LMI)
- Reduces region of attraction
  - A 'real' controller must provide *some* input in *every* circumstance
- Often unnecessary
  - Stable system, long horizon $\rightarrow$ will be stable and feasible in a (large) neighbourhood of the origin

Choice of Terminal Set and Cost: Summary

- Terminal constraint provides a sufficient condition for stability
- Region of attraction without terminal constraint may be larger than for MPC with terminal constraint but characterization of region of attraction extremely difficult
- $\mathcal{X}_f = 0$ simplest choice but small region of attaction for small N
- Solution for linear systems with quadratic cost
- In practice: Enlarge horizon and check stability by sampling
- With larger horizon length N, region of attraction approaches maximum control invariant set

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 10: Practical Issues

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Francesco Borrelli, UC Berkeley
Prof. Colin Jones, EPFL
Prof. Melanie Zeilinger, ETH Zurich

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 12.6-12.7].

Outline

1. Reference Tracking

2. Soft Constraints

3. Generalizing the Problem

Outline

1. Reference Tracking

2. Soft Constraints

3. Generalizing the Problem

Tracking problem

Consider the linear system model

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k
\end{aligned}
$$

Goal: Track given reference r such that $y_k \to r$ as $k \to \infty$.

Determine the steady state target condition x_s, u_s:

$$
\begin{aligned}
x_s &= Ax_s + Bu_s \\
Cx_s &= r
\end{aligned}
\quad \iff \quad
\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}
$$

Steady-state Target Problem

- In the presence of constraints: (x_s, u_s) has to satisfy state and input constraints.
- In case of multiple feasible u_s, compute 'cheapest' steady-state (x_s, u_s) corresponding to reference r:

$$\begin{aligned} \min \; & u_s^T R_s u_s \\ \text{subj. to } & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

- In general, we assume that the target problem is feasible
- If no solution exists: compute reachable set point that is 'closest' to r:

$$\begin{aligned} \min \; & (Cx_s - r)^T Q_s (Cx_s - r) \\ \text{subj. to } & x_s = Ax_s + Bu_s \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

RHC Reference Tracking

We now use control (MPC) to bring the system to a desired steady-state condition (x_s, u_s) yielding the desired output $y_k \to r$.

The MPC is designed as follows

$$
\begin{aligned}
\min_{u_0,\dots,u_{N-1}} \quad & \|y_N - Cx_s\|_P^2 + \sum_{k=0}^{N-1} \|y_k - Cx_s\|_Q^2 + \|u_k - u_s\|_R^2 \\
\text{subj. to} \quad & \text{[model constraints]} \\
& x_0 = x(k)
\end{aligned}
$$

Drawback: controller will show **offset** in case of unknown model error or disturbances.

RHC Reference Tracking without Offset (1/6)

Discrete-time, time-invariant system (possibly nonlinear, uncertain)

$$x_m(k+1) = g(x_m(k), u(k))$$
$$y_m(k) = h(x_m(k))$$

Objective:

- Design an RHC in order to make $y(k)$ track the reference signal $r(k)$, i.e., $(y(k) - r(k)) \to 0$ for $t \to \infty$.
- In the rest of the section we study step references and focus on zero steady-state tracking error, $y(k) \to r_\infty$ as $k \to \infty$.

Consider augmented model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d d(k) \\ d(k+1) &= d(k) \\ y(k) &= Cx(k) + C_d d(k) \end{aligned}$$

with constant disturbance $d(k) \in \mathbb{R}^{n_d}$.

RHC Reference Tracking without Offset (2/6)

State observer for augmented model

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(-y_m(k) + C\hat{x}(k) + C_d\hat{d}(k)\right)$$

Lemma

Suppose the observer is stable and the number of outputs p equals the dimension of the constant disturbance n_d. The observer steady state satisfies:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d\hat{d}_\infty \\ y_{m,\infty} - C_d\hat{d}_\infty \end{bmatrix}$$

where $y_{m,\infty}$ and u_∞ are the steady state measured outputs and inputs.

$\Rightarrow$ Observer output $C\hat{x}_\infty + C_d\hat{d}_\infty$ tracks the measurement $y_{m,\infty}$ without offset.

RHC Reference Tracking without Offset (3/6)

For offset-free tracking at steady state we want $y_{m,\infty} = r_\infty$.

The observer condition

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ y_{m,\infty} - C_d \hat{d}_\infty \end{bmatrix}$$

suggests that at steady state the MPC should satisfy

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{\text{target},\infty} \\ u_{\text{target},\infty} \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ r_\infty - C_d \hat{d}_\infty \end{bmatrix}$$

RHC Reference Tracking without Offset (4/6)

Formulate the RHC problem

$$
\begin{aligned}
\min_{U} \ & \|x_N - \bar{x}_k\|_P^2 + \sum_{k=0}^{N-1} \|x_k - \bar{x}_k\|_Q^2 + \|u_k - \bar{u}_t\|_R^2 \\
\text{subj. to } & x_{k+1} = Ax_k + Bu_k + B_d d_k, \quad && k = 0, \ldots, N \\
& x_k \in \mathcal{X},\ u_k \in \mathcal{U}, && k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f \\
& d_{k+1} = d_k, && k = 0, \ldots, N \\
& x_0 = \hat{x}(k) \\
& d_0 = \hat{d}(k),
\end{aligned}
$$

with the targets $\bar{u}_k$ and $\bar{x}_k$ given by

$$
\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}(k) \\ r(k) - C_d \hat{d}(k) \end{bmatrix}
$$

RHC Reference Tracking without Offset (5/6)

Denote by $\kappa(\hat{x}(k), \hat{d}(k), r(k)) = u_0^\star$ the control law when the estimated state and disturbance are $\hat{x}(k)$ and $\hat{d}(k)$, respectively.

Theorem

Consider the case where the number of constant disturbances equals the number of (tracked) outputs $n_d = p = r$. Assume the RHC is recursively feasible and unconstrained for $k \geq j$ with $j \in \mathbb{N}^+$ and the closed-loop system

$$
\begin{aligned}
x(k+1) &= f(x(k), \kappa(\hat{x}(k), \hat{d}(k), r(k))) \\
\hat{x}(k+1) &= (A + L_x C)\hat{x}(k) + (B_d + L_x C_d)\hat{d}(k) \\
&\quad + B\kappa(\hat{x}(k), \hat{d}(k), r(k)) - L_x y_m(k) \\
\hat{d}(k+1) &= L_d C\hat{x}(k) + (I + L_d C_d)\hat{d}(k) - L_d y_m(k)
\end{aligned}
$$

converges to $\hat{x}(k) \to \hat{x}_\infty$, $\hat{d}(k) \to \hat{d}_\infty$, $y_m(k) \to y_{m,\infty}$ as $t \to \infty$.

Then $y_m(k) \to r_\infty$ as $t \to \infty$.

RHC Reference Tracking without Offset (6/6)

Question: How do we choose the matrices B_d and C_d in the augmented model?

Lemma

The augmented system, with the number of outputs p equal to the dimension of the constant disturbance n_d, and $C_d = I$ is observable if and only if (C, A) is observable and

$$\det \begin{bmatrix} A - I & B_d \\ C & I \end{bmatrix} = \det(A - I - B_d C) \neq 0.$$

Remark: If the plant has no integrators, then $\det(A - I) \neq 0$ and we can choose $B_d = 0$. If the plant has integrators then B_d has to be chosen specifically to make $\det(A - I - B_d C) \neq 0$.

Outline

Soft Constraints: Motivation

- Input constraints are dictated by physical constraints on the actuators and are usually “hard”
- State/output constraints arise from practical restrictions on the allowed operating range and are **rarely hard**
- Hard state/output constraints always lead to **complications in the controller implementation**
  - Feasible operating regime is constrained even for stable systems
  - Controller patches must be implemented to generate reasonable control action when measured/estimated states move outside feasible range because of disturbances or noise
- In industrial implementations, typically, state constraints are **softened**

Mathematical Formulation

- **Original** problem:

$$\min_{z} f(z)$$
$$\text{subj. to } g(z) \leq 0$$

  Assume for now $g(z)$ is scalar valued.

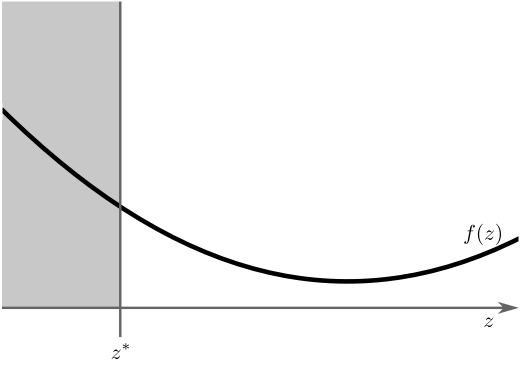
- **"Softened"** problem:

$$\min_{z,\epsilon} f(z) + l(\epsilon)$$
$$\text{subj. to } g(z) \leq \epsilon$$
$$\epsilon \geq 0$$

Requirement on $l(\epsilon)$
If the original problem has a feasible solution $z^{\star}$, then the softened problem should have the same solution $z^{\star}$, and $\epsilon = 0$.

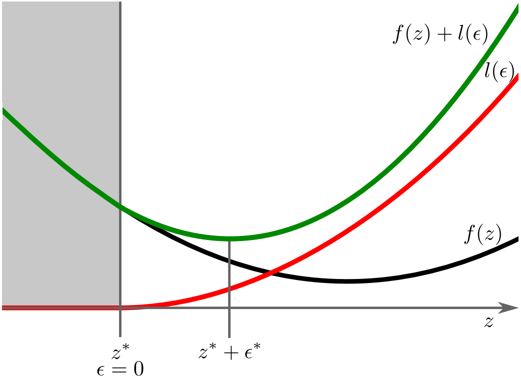
Note: $l(\epsilon) = v \cdot \epsilon^2$ does not meet this requirement for any $v > 0$ as demonstrated next.

Quadratic Penalty



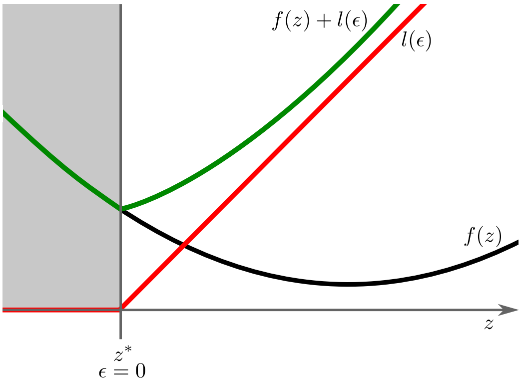
- Constraint function $g(z) \triangleq z - z^\star \leq 0$ induces feasible region (grey) $\Longrightarrow$ minimizer of the original problem is $z^\star$

Quadratic Penalty



- Constraint function $g(z) \triangleq z - z^\star \leq 0$ induces feasible region (grey)
  $\Longrightarrow$ minimizer of the original problem is $z^\star$
- **Quadratic penalty** $l(\epsilon) = v \cdot \epsilon^2$ for $\epsilon \geq 0$
  $\Longrightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^\star + \epsilon^\star, \epsilon^\star)$ instead of $(z^\star, 0)$

Linear Penalty



- Constraint function $g(z) := z - z^\star \leq 0$ induces feasible region (grey)
  $\Longrightarrow$ minimizer of the original problem is $z^\star$
- **Linear penalty** $l(\epsilon) = u \cdot \epsilon$ for $\epsilon \geq 0$ with u chosen large enough so that $u + \lim_{z \to z^\star} f'(z) > 0$
  $\Longrightarrow$ minimizer of $f(z) + l(\epsilon)$ is $(z^\star, 0)$

Main Result

Theorem: Exact Penalty Function

$l(\epsilon) = u \cdot \epsilon$ satisfies the requirement for any $u > u^\star \geq 0$, where $u^\star$ is the optimal Lagrange multiplier for the original problem.

- **Disadvantage:** $l(\epsilon) = u \cdot \epsilon$ renders the cost non-smooth.
- Therefore in practice, to get a smooth penalty, we use

$$l(\epsilon) = u \cdot \epsilon + v \cdot \epsilon^2$$

  with $u > u^\star$ and $v > 0$.
- Extension to multiple constraints $g_j(z) \leq 0, \ j = 1, \ldots, r$:

$$l(\epsilon) = \sum_{j=1}^{r} u_j \cdot \epsilon_j + v_j \cdot \epsilon_j^2 \qquad (1)$$

  where $u_j > u_j^\star$ and $v_j > 0$ can be used to weight violations (if necessary) differently.

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 11: Explicit MPC

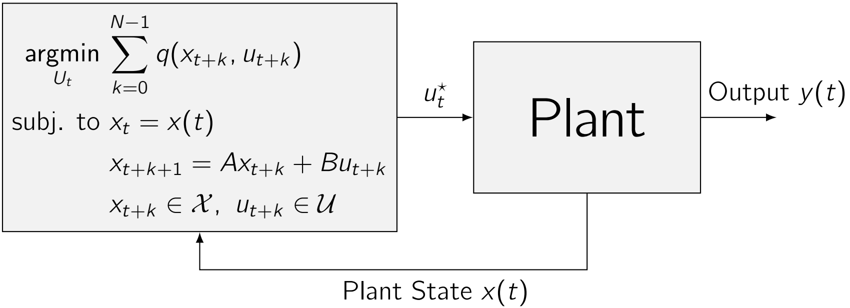
Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Francesco Borrelli, UC Berkeley
Prof. Colin Jones, EPFL

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 11].

Introduction



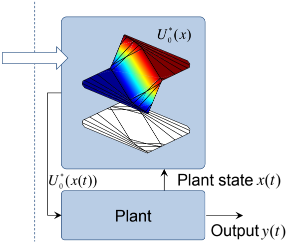
- Requires at each time step on-line solution of an optimization problem

Introduction

OFFLINE

ONLINE

$$
\begin{aligned}
U_0^*(x(t)) = \text{argmin } & x_N^T P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\
\text{subj. to } & x_0 = x(t) \\
& x_{k+1} = Ax_k + Bu_k, \;\; k = 0, \ldots, N-1 \\
& x_k \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \;\; k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f
\end{aligned}
$$



- Optimization problem is parameterized by state
- Pre-compute control law as function of state x
- Control law is piecewise affine for linear system/constraints

Result: Online computation dramatically reduced and **real-time**
Tool: **Parametric programming**

mpQP - Problem formulation

$$
\begin{aligned}
J^*(x) = \min_z \quad & \tfrac{1}{2} z' H z, \\
\text{subj. to} \quad & Gz \leq w + Sx
\end{aligned}
$$

where $H > 0$, $z \in \mathbb{R}^s$, $x \in \mathbb{R}^n$ and $G \in \mathbb{R}^{m \times s}$.

Given a closed and bounded polyhedral set $\mathcal{K} \subset \mathbb{R}^n$ of parameters denote by $\mathcal{K}^* \subseteq \mathcal{K}$ the region of parameters $x \in \mathcal{K}$ such that the problem is feasible

$$
\mathcal{K}^* := \{x \in K : \exists z,\ Gz \leq w + Sx\}
$$

Goals:

1. find $z^*(x) = \text{argmin}_z\ J(z, x)$,
2. find all x for which the problem has a solution
3. compute the value function $J^*(x)$

Active Set and Critical Region

Let $I := \{1, \ldots, m\}$ be the set of constraint indices.

Definition: Active Set

We define the active set at x, $A(x)$, and its complement, $NA(x)$, as

$$A(x) := \{i \in I : G_i z^*(x) - S_i x = w_i\}$$
$$NA(x) := \{i \in I : \; G_i z^*(x) - S_i x < w_i\}.$$

G_i, S_i and w_i are the i-th row of G, S and w, respectively.

Definition: Critical Region

CR_A is the set of parameters x for which the same set $A \subseteq I$ of constraints is active at the optimum. For a given $\bar{x} \in \mathcal{K}^*$ let $(A, NA) := (A(\bar{x}), NA(\bar{x}))$. Then,

$$CR_A := \{x \in \mathcal{K}^* : A(x) = A\}.$$

mpQP - Global properties of the solution

The following theorem summarizes the properties of the mpQP solution.

Theorem: Solution of mpQP
i) The feasible set $\mathcal{K}^*$ is a **polyhedron**. ii) The optimizer function $z^*(x) : \mathcal{K}^* \to \mathbb{R}^m$ is: • **continuous** • **polyhedral piecewise affine over** $\mathcal{K}^*$. It is affine in each critical region $\mathcal{CR}_i$, every $\mathcal{CR}_i$ is a polyhedron and $\bigcup \mathcal{CR}_i = \mathcal{K}^*$. iii) The value function $J^*(x) : \mathcal{K}^* \to \mathbb{R}$ is: • **continuous** • **convex** • **polyhedral piecewise quadratic over** $\mathcal{K}^*$, it is quadratic in each $\mathcal{CR}_i$

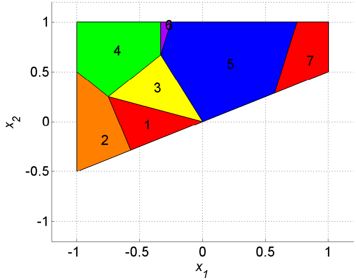
mpQP - Example (1/4)

Consider the example

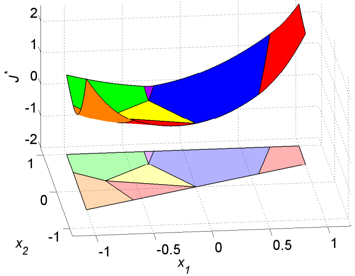
$$
\begin{aligned}
\min_{z(x)} \quad & \tfrac{1}{2}(z_1^2 + z_2^2) \\
\text{subj. to} \quad & z_1 \le 1 + x_1 + x_2 \\
& -z_1 \le 1 - x_1 - x_2 \\
& z_2 \le 1 + x_1 - x_2 \\
& -z_2 \le 1 - x_1 + x_2 \\
& z_1 - z_2 \le x_1 + 3x_2 \\
& -z_1 + z_2 \le -2x_1 - x_2 \\
& -1 \le x_1 \le 1, \ -1 \le x_2 \le 1
\end{aligned}
$$

mpQP - Example (2/4)

The explicit solution is defined over $i = 1, \dots, 7$ regions $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \le b_i\}$ in the parameter space $x_1 - x_2$.



Critical regions



Piecewise quadratic objective function $J^*(x)$

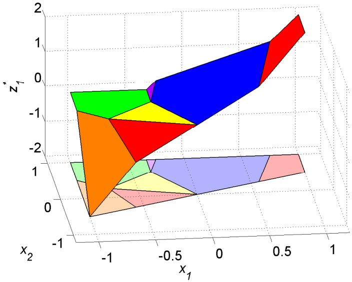
mpQP - Example (3/4)

Primal solution is given as piecewise affine function $z(x) = F_i + g_i x$ if $x \in \mathcal{P}_i$.

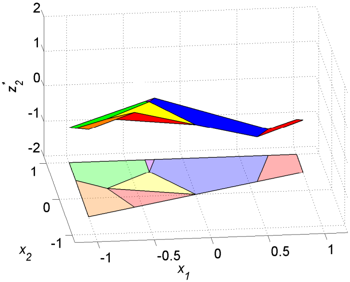
$$z^*(x) = \begin{cases} \begin{pmatrix} 0.5 & 1.5 \\ -0.5 & -1.5 \end{pmatrix} x \text{ if } x \in \mathcal{P}_1 \\ \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{if } x \in \mathcal{P}_2 \\ \vdots \\ \vdots \end{cases}$$

mpQP - Example (4/4)

Primal solution is given as piecewise affine function $z(x) = F_i + g_i x$ if $x \in \mathcal{P}_i$.



Piecewise affine function $z_1^*(x)$



Piecewise affine function $z_2^*(x)$

2-Norm State Feedback Solution

Main Results

1. The **Open loop optimal control function** can be obtained by solving the mp-QP problem and calculating $U_0^*(x(0)),\ \forall x(0) \in \mathcal{X}_0$ as $U_0{}^* = z^*(x(0)) - H^{-1}F'x(0)$.
2. The first component of the multiparametric solution has the form

$$u^*(0) = f_0(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

$f_0 : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and piecewise affine on polyhedra

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \dots, N_0^r$$

3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \dots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. The value function $J_0^*(x(0))$ is convex and piecewise quadratic on polyhedra.

Example

Consider the double integrator

$$\begin{cases} x(t+1) & = & \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) & = & \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

subject to constraints

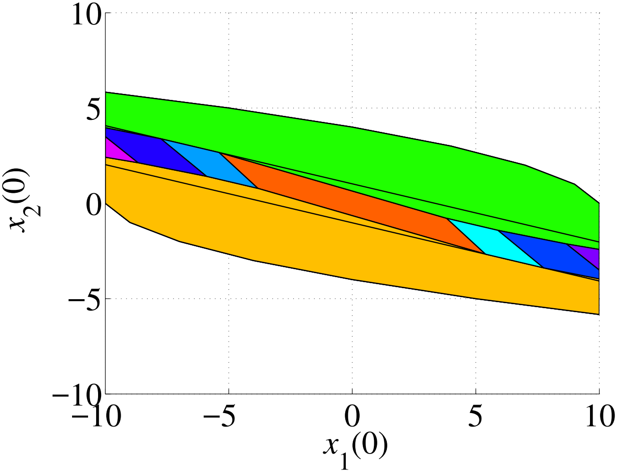
$$-1 \leq u(k) \leq 1,\ k = 0, \dots, 5$$

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix},\ k = 0, \dots, 5$$

Compute the **state feedback** optimal controller $u^*(0)(x(0))$ solving the

CFTOC problem with $N = 6$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.1$, P the solution of the ARE, $\mathcal{X}_f = \mathbb{R}^2$.

Example

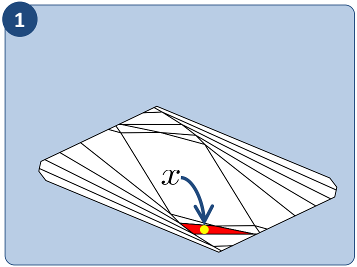


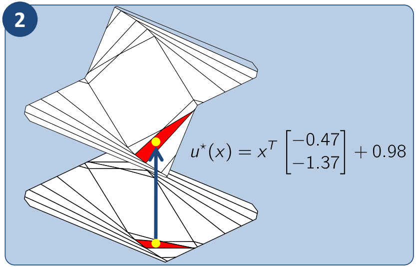
Partition of state space for the piecewise affine control law $u^*(0)$ ($N_0^r = 13$)

Online evaluation: Point location

Calculation of piecewise affine function:

1. Point location
2. Evaluation of affine function





University of Pennsylvania, ESE619

Model Predictive Control

Chapter 12: Hybrid MPC

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Francesco Borrelli, UC Berkeley
Prof. Colin Jones, EPFL

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 16, 17].

Introduction

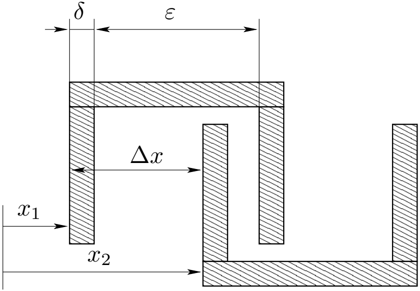
Up to this point: Discrete-time linear systems with linear constraints.

We now consider MPC for systems with

1. **Continuous dynamics**: described by one or more difference (or differential) equations; states are continuous-valued.
2. **Discrete events**: state variables assume **discrete** values, e.g.
   - binary digits $\{0, 1\}$,
   - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, . . .
   - finite set of symbols

Hybrid systems: Dynamical systems whose state evolution depends on an interaction between continuous dynamics and discrete events.

Mechanical System with Backlash

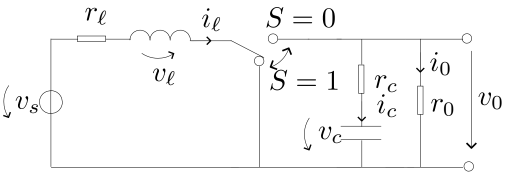


- **Continuous dynamics**: states x_1, x_2, $\dot{x}_1$, $\dot{x}_2$.
- **Discrete events**:
  a) *"contact mode"* $\Rightarrow$ mechanical parts are in contact and the force is transmitted. Condition:

$$[(\Delta x = \delta) \wedge (\dot{x}_1 > \dot{x}_2)] \bigvee [(\Delta x = \varepsilon) \wedge (\dot{x}_2 > \dot{x}_1)]$$

  b) *"backlash mode"* $\Rightarrow$ mechanical parts are not in contact

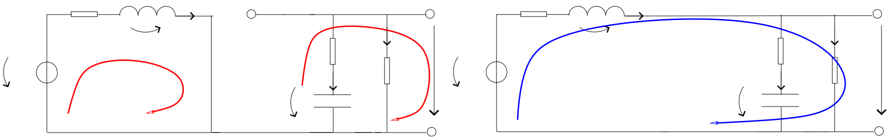
DCDC Converter



- **Continuous dynamics**: states v_ℓ, i_ℓ, v_c, i_c, v_0, i_0
- **Discrete events**: $S=0$, $S=1$

Mode 1 ($S=1$)

Mode 2 ($S=0$)



Piecewise Affine (PWA) Systems

PWA systems are defined by:

- **affine dynamics and output** in each region:

$$\left.\begin{cases} x(t+1) &= A_i x(t) + B_i u(t) + f_i \\ y(t) &= C_i x(t) + D_i u(t) + g_i \end{cases}\right\} \text{ if } (x(t), u(t)) \in \mathcal{X}_{i(t)}$$

- **polyhedral partition** of the (x, u)-space:

$$\{\mathcal{X}_i\}_{i=1}^{s} := \{x, u \mid H_i x + J_i u \leq K_i\}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$

Physical constraints on $x(t)$ and $u(t)$ are defined by polyhedra $\mathcal{X}_i$

Piecewise Affine (PWA) Systems

Examples:

- *linearization* of a non-linear system at different operating point $\Rightarrow$ useful as an approximation tool
- *closed-loop MPC system* for linear constrained systems
- When the mode i is an exogenous variable, the partition disappears and we refer to the system as a **Switched Affine System** (SAS)

Definition: Well-Posedness

Let P be a PWA system and let $\mathcal{X} = \cup_{i=1}^{s} \mathcal{X}_i \subseteq \mathbb{R}^{n+m}$ be the polyhedral partition associated with it. System P is called **well-posed** if for all pairs $(x(t), u(t)) \in \mathcal{X}$ *there exists only one* index $i(t)$ satisfying the membership condition.

Binary States, Inputs, and Outputs

Remark: In the previous example, the PWA system has only continuous states and inputs.

We will formulate PWA systems including binary state and inputs by treating 0–1 binary variables as:

- **Numbers**, over which arithmetic operations are defined,
- **Boolean variables**, over which Boolean functions are defined.

We will use the notation $x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix} \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $n := n_c + n_\ell$; $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$, $p := p_c + p_\ell$; $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$, $m := m_c + m_\ell$.

Boolean Algebra: Basic Definitions and Notation

- **Boolean variable:** A variable δ is a Boolean variable if $\delta \in \{0, 1\}$, where "$\delta = 0$" means "false", "$\delta = 1$" means "true".

- **A Boolean expression** is obtained by combining Boolean variables through the logic operators $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\leftarrow$ (implied by), $\rightarrow$ (implies), and $\leftrightarrow$ (iff).

- **A Boolean function** $f : \{0, 1\}^{n-1} \mapsto \{0, 1\}$ is used to define a Boolean variable δ_n as a logic function of other variables $\delta_1, \ldots, \delta_{n-1}$:

$$\delta_n = f(\delta_1, \delta_2, \ldots, \delta_{n-1}).$$

Mixed Logical Dynamical Systems

Goal: Describe hybrid system in form compatible with optimization software:

- continuous and Boolean variables
- linear equalities and inequalities

Idea: associate to each Boolean variable p_i a binary integer variable δ_i:

$$p_i \Leftrightarrow \{\delta_i = 1\}, \quad \neg p_i \Leftrightarrow \{\delta_i = 0\}$$

and embed them into a set of constraints as **linear integer inequalities**.

Two main steps:

1. Translation of Logic Rules into Linear Integer Inequalities
2. Translation continuous and logical components into Linear Mixed-Integer Relations

Final result: a compact model with linear equalities and inequalities involving real and binary variables

MLD Hybrid Model

A DHA can be converted into the following MLD model

$$
\begin{aligned}
x_{t+1} &= Ax_t + B_1 u_t + B_2 \delta_t + B_3 z_t \\
y_t &= Cx_t + D_1 u_t + D_2 \delta_t + D_3 z_t \\
E_2 \delta_t + E_3 z_t &\leq E_4 x_t + E_1 u_t + E_5
\end{aligned}
$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$, $\delta \in \{0,1\}^{r_\ell}$ and $z \in \mathbb{R}^{r_c}$.

Physical constraints on continuous variables:

$$
\mathcal{C} = \left\{ \begin{bmatrix} x_c \\ u_c \end{bmatrix} \in \mathbb{R}^{n_c+m_c} \;\middle|\; Fx_c + Gu_c \leq H \right\}
$$

HYbrid System DEscription Language

HYSDEL

- based on DHA
- enables description of discrete-time hybrid systems in a compact way:
  - automata and propositional logic
  - continuous dynamics
  - A/D and D/A conversion
  - definition of constraints
- automatically **generates MLD models** for MATLAB
- freely available from:

`http://control.ee.ethz.ch/~hybrid/hysdel/`

Optimal Control for Hybrid Systems: General Formulation

Consider the CFTOC problem:

$$J^*(x(t)) = \min_{U_0} \ p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k, \delta_k, z_k),$$

$$\text{s.t.} \begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ \\ E_2 \delta_k + E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \\ \\ x_N \in \mathcal{X}_f \\ x_0 = x(t) \end{cases}$$

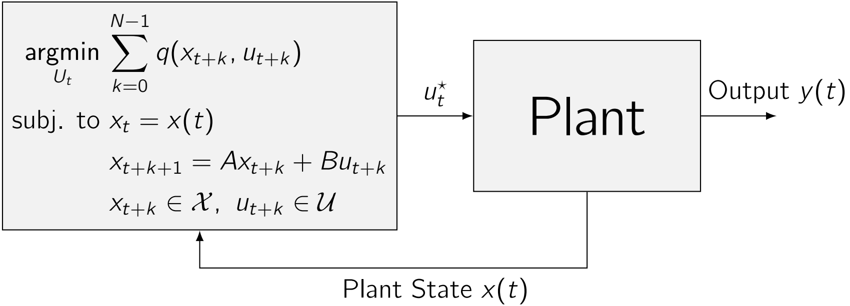
where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$ and $z \in \mathbb{R}^{r_c}$ and

$$U_0 = \{u_0, u_1, \ldots, u_{N-1}\}$$

Mixed Integer Optimization

Model Predictive Control of Hybrid Systems

MPC solution: Optimization in the loop



As for linear MPC, at each sample time:

- Measure / estimate current state $x(t)$
- Find the optimal input sequence for the entire planning window N: $U_t^* = \{u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*\}$
- Implement only the **first** control action u_t^*
- **Key difference: Requires online solution of an MILP or MIQP**

Summary

- Hybrid systems: mixture of continuous and discrete dynamics
  - Many important systems fall in this class
  - Many tricks involved in modeling - automatic systems available to convert to consistent form

- Optimization problem becomes a mixed-integer linear / quadratic program
  - NP-hard (exponential time to solve)
  - Advanced commercial solvers available

- MPC theory (invariance, stability, etc) applies
  - Computing invariant sets is usually extremely difficult
  - Computing the optimal solution is extremely difficult (sub-optimal ok)

University of Pennsylvania, ESE619

Model Predictive Control

Chapter 13: Robust MPC

Prof. Manfred Morari

Spring 2019

Coauthors: Prof. Colin Jones, EPFL
Prof. Melanie Zeilinger, ETH Zurich

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems,
Cambridge University Press, 2017. [Ch. 15].

Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise

3. Robust Open-Loop MPC

4. Closed-Loop Predictions

5. Tube-MPC

6. Nominal MPC with noise

Lecture Take Homes

1. MPC relies on a model, but models are far from perfect

2. Noise and model inaccuracies can cause:
    - Constraint violation
    - Sub-optimal behaviour can result

3. Persistent noise prevents the system from converging to a single point

4. Can incorporate some noise models into the MPC formulation
    - Solving the resulting optimal control problem is extremely difficult
    - Many approximations exist, but most are very conservative

Examples of Common Uncertainty Models

Additive Bounded Noise

$$g(x, u, w; \theta) = Ax + Bu + w \ , \quad w \in \mathbb{W}$$

A, B known, w unknown and changing with each sample

- Dynamics are linear, but impacted by random, bounded noise at each time step
- Can model many nonlinearities in this fashion, but often a conservative model
- The noise is *persistent*, i.e., it does not converge to zero in the limit

The next lectures will focus on uncertainty models of this form.

Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise

3. Robust Open-Loop MPC

4. Closed-Loop Predictions

5. Tube-MPC

6. Nominal MPC with noise

Goals of Robust Constrained Control

Uncertain constrained linear system		
$x^+ = Ax + Bu + w$	$(x, u) \in \mathcal{X}, \mathcal{U}$	$w \in \mathbb{W}$

Design control law $u = \kappa(x)$ such that the system:

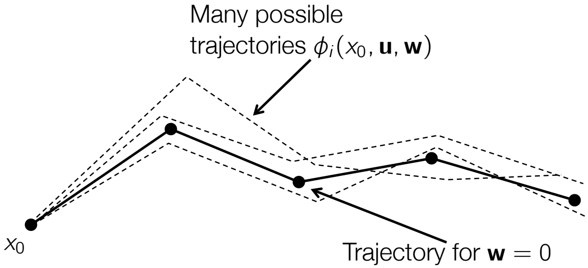
1. Satifies constraints : $\{x_i\} \subset \mathcal{X}$, $\{u_i\} \subset \mathcal{U}$ for all disturbance realizations
2. Is stable: Converges to a neighbourhood of the origin
3. Optimizes (expected/worst-case) "performance"
4. Maximizes the set $\{x_0 \,|\, \text{Conditions 1-3 are met}\}$

Challenge: Cannot predict where the state of the system will evolve
We can only compute a set of trajectories that the system *may* follow

Idea: Design a control law that will satisfy constraints and stabilize the system *for all possible disturbances*

Uncertain State Evolution

Given the current state x_0, the model $x^+ = Ax + Bu + w$ and the set $\mathbb{W}$, where can the state be i steps in the future?



Define $\phi_i(x_0, \vec{u}, \vec{w})$ as the state that the system will be in at time i if the state at time zero is x_0, we apply the input $\vec{u} := \{u_0, \ldots, u_{N-1}\}$ and we observe the disturbance $\vec{w} := \{w_0, \ldots, w_{N-1}\}$.

Uncertain State Evolution

Nominal system

$$x^+ = Ax + Bu$$

$$x_1 = Ax_0 + Bu_0$$
$$x_2 = A^2x_0 + ABu_0 + Bu_1$$
$$\vdots$$
$$x_i = A^i x_0 + \sum_{k=0}^{i-1} A^k B u_{i-k}$$

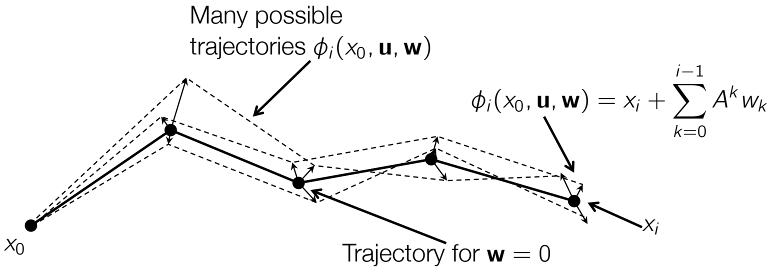
Uncertain system

$$x^+ = Ax + Bu + w \;, w \in \mathbb{W}$$

$$\phi_1 = Ax_0 + Bu_0 + w_0$$
$$\phi_2 = A^2x_0 + ABu_0 + Bu_1 + Aw_0 + w_1$$
$$\vdots$$
$$\phi_i = A^i x_0 + \sum_{k=0}^{i-1} A^k B u_{i-k} + \sum_{k=0}^{i-1} A^k w_{i-k}$$
$$\phi_i = x_i + \sum_{k=0}^{i-1} A^k w_{i-k}$$

Uncertain evolution is the nominal system + offset caused by the disturbance (Follows from linearity)

Uncertain State Evolution



Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise

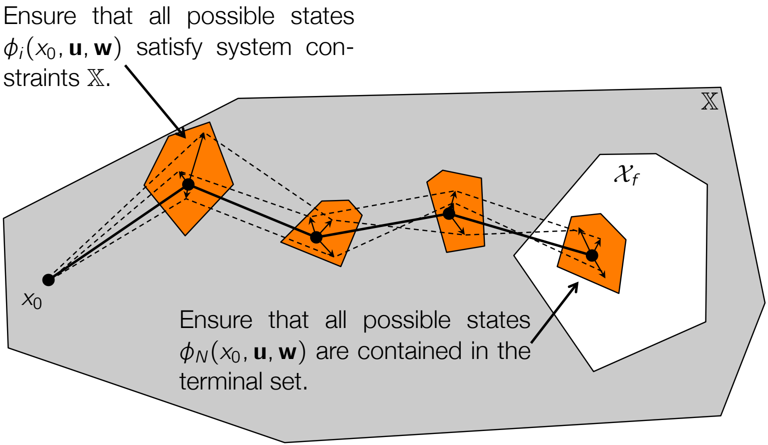
3. Robust Open-Loop MPC

4. Closed-Loop Predictions

5. Tube-MPC

6. Nominal MPC with noise

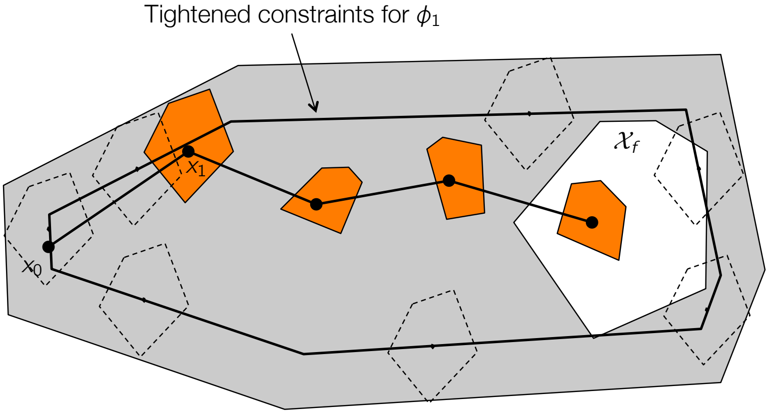
Robust Constraint Satisfaction



The idea: Compute a set of tighter constraints such that if **the nominal system** meets these constraints, then the uncertain system will too.
We then do MPC **on the nominal system.**

Robust Constraint Satisfaction

Goal: Ensure that constraints are satisfied for the MPC sequence.



Require: $x_i \in \mathcal{X} \ominus \begin{bmatrix} I & A^0 & \dots & A^{i-1} \end{bmatrix} \mathbb{W}^i$ and

Nominal x_i satisfies tighter constraints $\rightarrow$ Uncertain state does too

Putting it Together

Robust Open-Loop MPC

$$\min_{\vec{u}} \sum_{i=0}^{N-1} l(x_i, u_i) + V_f(x_N)$$
$$\text{subj. to } x_{i+1} = Ax_i + Bu_i$$
$$x_i \in \mathcal{X} \ominus \mathcal{A}_i \mathbb{W}^i$$
$$u_i \in \mathcal{U}$$
$$x_N \in \tilde{\mathcal{X}}_f$$

where $\mathcal{A}_i := \begin{bmatrix} A^0 & A^1 & \dots & A^i \end{bmatrix}$ and $\tilde{\mathcal{X}}_f$ is a robust invariant set for the system $x^+ = (A + BK)x$ for some stabilizing K.

We do **nominal MPC**, but with tighter constraints on the states and inputs.

We can be sure that if the nominal system satisfies the tighter constraints, then the uncertain system will satisfy the real constraints.

$\Rightarrow$ Downside is that $\mathcal{A}^i \mathbb{W}^i$ can be very large

Outline

MPC as a Game

Two players: Controller vs Disturbance

$$x^+ = f(x, u) + w$$

1. Controller chooses his move u
2. Disturbance decides on his move w **after seeing the controller's move**

MPC as a Game

Two players: Controller vs Disturbance

$$x^+ = f(x, u) + w$$

1. Controller chooses his move u
2. Disturbance decides on his move w **after seeing the controller's move**

What are we assuming when making robust predictions?

1. Controller chooses a **sequence** of N moves in the future $\{u_0, \ldots, u_{N-1}\}$
2. Disturbance chooses N moves **knowing all N moves of the controller**

We are assuming that the controller will do the same thing in the future no matter what the disturbance does!

Can we do better?

Closed-Loop Predictions

What should the future prediction look like?

1. Controller decides his first move u_0

2. Disturbance chooses his first move w_0

3. Controller decides his second move $u_1(x_1)$ **as a function of the first disturbance** w_0 **(recall** $x_1 = Ax_0 + Bu_0 + w_0$**)**

4. Disturbance chooses his second move w_1 as a function of u_1

5. Controller decides his second move $u_2(x_2)$ **as a function of the first two disturbances** w_0**,** w_1

6. ...

Closed-Loop Predictions

We want to optimize over a **sequence of functions** $\{u_0, \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$, where $\mu_i(x_i) : \mathbb{R}^n \to \mathbb{R}^m$ is called a **control policy**, and maps the state at time i to an input at time i.

Notes:

- This is the same as making μ a function of the disturbances to time i, since the state is a function of the disturbances up to that point
- The first input u_0 is a function of the current state, which is known. Therefore it is not a function, but a single value.

The problem: We can't optimize over arbitrary functions!

Closed-Loop MPC

A solution: Assume some structure on the functions μ_i

Pre-stabilization $\mu_i(x) = Kx + v_i$

- Fixed K, such that $A + BK$ is stable
- Simple, often conservative

Linear feedback $\mu_i(x) = K_i x + v_i$

- Optimize over K_i and v_i
- Non-convex. Extremely difficult to solve...

Disturbance feedback $\mu_i(x) = \sum_{j=0}^{i-1} M_{ij} w_j + v_i$

- Optimize over M_{ij} and v_i
- Equivalent to linear feedback, but convex!
- Can be very effective, but computationally intense.

Tube-MPC $\mu_i(x) = v_i + K(x - \bar{x}_i)$

- Fixed K, such that $A + BK$ is stable
- Optimize over $\bar{x}_i$ and v_i
- Simple, and can be effective

We will cover tube-MPC in this lecture.

Outline

1. Uncertainty Models
2. Impact of Bounded Additive Noise
3. Robust Open-Loop MPC
4. Closed-Loop Predictions
5. Tube-MPC
6. Nominal MPC with noise

Tube-MPC

$$x^+ = Ax + Bu + w \qquad (x, u) \in \mathcal{X} \times \mathcal{U} \qquad w \in \mathbb{W}$$

The idea: Seperate the available control authority into two parts

1. A portion that steers the noise-free system to the origin $z^+ = Az + Bv$
2. A portion that compensates for deviations from this system $e^+ = (A + BK)e + w$

We fix the linear feedback controller K offline, and optimize over the nominal trajectory $\{v_0, \ldots, v_{N-1}\}$, which results in a convex problem.

[0] Further reading: D.Q. Mayne, M.M. Seron and S.V. Rakovic, Robust model predictive control of constrained linear systems with bounded disturbances, Automatica, Volume 41, Issue 2, February 2005

System Decomposition

Define a 'nominal', noise-free system:

$$z_{i+1} = Az_i + Bv_i$$

Define a 'tracking' controller, to keep the real trajectory close to the nominal

$$u_i = K(x_i - z_i) + v_i$$

for some linear controller K, which stabilizes the nominal system.

Define the error $e_i = x_i - z_i$, which gives the error dynamics:

$$\begin{aligned}
e_{i+1} &= x_{i+1} - z_{i+1} \\
&= Ax_i + Bu_i + w_i - Az_i - Bv_i \\
&= Ax_i + BK(x_i - z_i) + Bv_i + w_i - Az_i - Bv_i \\
&= (A + BK)(x_i - z_i) + w_i \\
&= (A + BK)e_i + w_i
\end{aligned}$$

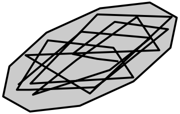
Error Dynamics

Bound maximum error, or how far the 'real' trajectory is from the nominal

$$e_{i+1} = (A + BK)e_i + w_i \qquad\qquad w_i \in \mathbb{W}$$

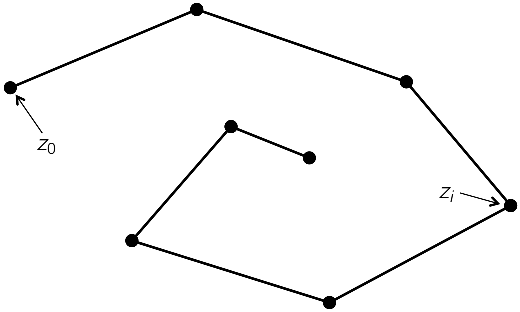
Dynamics $A + BK$ are stable, and the set $\mathbb{W}$ is bounded, so there is some set $\mathcal{E}$ that e will stay inside for all time.

We want the smallest such set (the 'minimal invariant set')



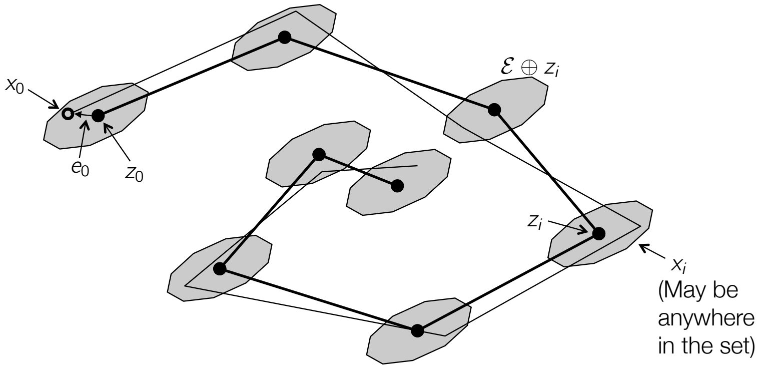
We will cover how to compute this set later

Tube-MPC : The Idea



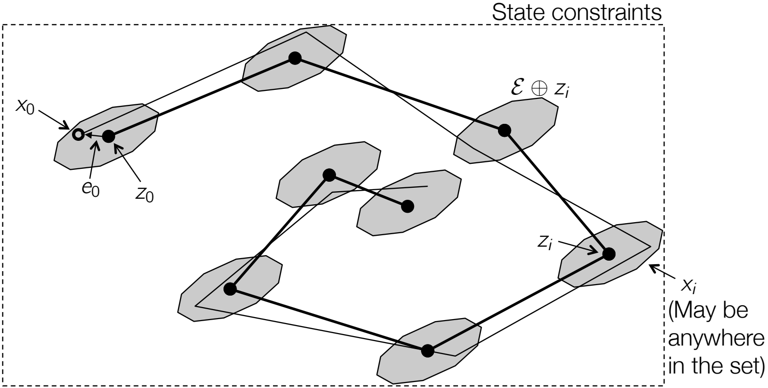
We want to ignore the noise and plan the **nominal trajectory**

Tube-MPC : The Idea



We know that the real trajectory stays 'nearby' the nominal one: $x_i \in z_i \oplus \mathcal{E}$ **because we plan to apply the controller** $u_i = K(x_i - z_i) + v_i$ **in the future** (we won't actually do this, but it's a valid sub-optimal plan)

Tube-MPC : The Idea



We must ensure that all possible state trajectories satisfy the constraints
This is now equivalent to ensuring that $z_i \oplus \mathcal{E} \subset \mathcal{X}$
(Satisfying input constraints is now more complex - more later)

Tube-MPC

What do we need to make this work?

- Compute the set $\mathcal{E}$ that the error will remain inside
- Modify constraints on nominal trajectory $\{z_i\}$ so that $z_i \oplus \mathcal{E} \subset \mathcal{X}$ and $v_i \in \mathcal{U} \ominus K\mathcal{E}$
- Formulate as convex optimization problem

…and then prove that

- Constraints are robustly satisfied
- The closed-loop system is robustly stable

Noisy System Trajectory

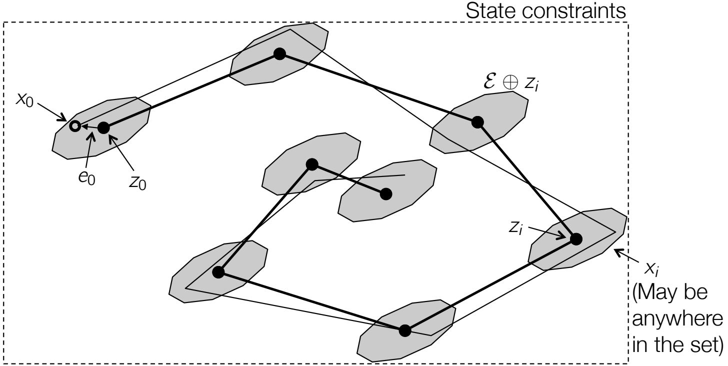
Given the nominal trajectory z_i, what can the noisy system trajectory do?

$$x_i = z_i + e_i$$

Don't know what error will be at time i, but it will be in the set $\mathcal{E}$

Therefore, x_i can only be up to $\mathcal{E}$ far from z_i

$$x_i \in z_i \oplus \mathcal{E} = \{z_i + e \,|\, e \in \mathcal{E}\}$$



Constraint Tightening

Goal: $(x_i, u_i) \in \mathcal{X} \times \mathcal{U}$ for all $\{w_0, \ldots, w_{i-1}\} \in \mathbb{W}^i$

We want to work with the nominal system $z^+ = Az + Bv$ but ensure that the noisy system $x^+ = Ax + Bu + w$ satisfies the constraints.

Sufficient condition:

$$z_i \oplus \mathcal{E} \subseteq \mathcal{X} \quad \Leftarrow \quad z_i \in \mathcal{X} \ominus \mathcal{E}$$

The set $\mathcal{E}$ is known offline - we can compute the constraints $\mathcal{X} \ominus \mathcal{E}$ offline!

A similar condition holds for the inputs:

$$u_i \in K\mathcal{E} \oplus v_i \subset \mathcal{U} \quad \Leftarrow \quad v_i \in \mathcal{U} \ominus K\mathcal{E}$$

Tube-MPC Problem Formulation

Tube-MPC

Feasible set: $$\mathcal{Z}(x_0) := \left\{ \vec{z}, \vec{v} \;\middle|\; \begin{array}{ll} z_{i+1} = Az_i + Bv_i & i \in [0,\ N-1] \\ z_i \in \mathcal{X} \ominus \mathcal{E} & i \in [0,\ N-1] \\ v_i \in \mathcal{U} \ominus K\mathcal{E} & i \in [0,\ N-1] \\ z_N \in \mathcal{X}_f & \\ x_0 \in z_0 \oplus \mathcal{E} & \end{array} \right\}$$

Cost function: $$V(\vec{z}, \vec{v}) := \sum_{i=0}^{N-1} l(z_i, v_i) + V_f(z_N)$$

Optimization problem: $$(\vec{v}^\star(x_0), \vec{z}^\star(x_0)) = \underset{\vec{v},\vec{z}}{\operatorname{argmin}} \left\{ V(\vec{z}, \vec{v}) \,\middle|\, (\vec{z}, \vec{v}) \in \mathcal{Z}(x_0) \right\}$$

Control law: $$\mu_{\text{tube}}(x) := K(x - z_0^\star(x)) + v_0^\star(x)$$

- Optimizing the nominal system, with tightened state an input constraints
- First tube center is optimization variable → has to be within $\mathcal{E}$ of x_0
- The cost is with respect to the tube centers
- The terminal set is with respect to the tightened constraints

Putting it all together: Tube MPC

To implement tube MPC:

— Offline —

1. Choose a stabilizing controller K so that $\|A + BK\| < 1$
2. Compute the minimal robust invariant set $\mathcal{E} = F_\infty$ for the system $x^+ = (A + BK)x + w$, $w \in \mathbb{W}$[1]
3. Compute the tightened constraints $\tilde{\mathcal{X}} := \mathcal{X} \ominus \mathcal{E}$, $\tilde{\mathcal{U}} := \mathcal{U} \ominus \mathcal{E}$
4. Choose terminal weight function V_f and constraint $\mathcal{X}_f$ satisfying assumptions on slide 88

— Online —

1. Measure / estimate state x
2. Solve the problem $(\vec{v}^\star(x), \vec{z}^\star(x)) = \operatorname{argmin}_{\vec{v},\vec{z}} \{V(\vec{z}, \vec{v}) \,|\, (\vec{z}, \vec{v}) \in \mathcal{Z}(x)\}$ (Slide 86)
3. Set the input to $u = K(x - z_0^\star(x)) + v_0^\star(x)$

[1] Note that it is often not possible to compute the minimal robust invariant set, as it may have an infinite number of facets. Therefore, we often take an invariant outer approximation.

Example

System dynamics

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u + w \qquad \mathbb{W} := \{w \,|\, |w_1| \leq 0.01,\ |w_2| \leq 0.1\}$$

Constraints:

$$\mathcal{X} := \{x \,|\, \|x\|_\infty \leq 1\} \qquad \mathcal{U} := \{u \,|\, \|u\| \leq 1\}$$

Stage cost is:

$$l(z, v) := {z_i}^\top Q z_i + {v_i}^\top R v_i$$

where

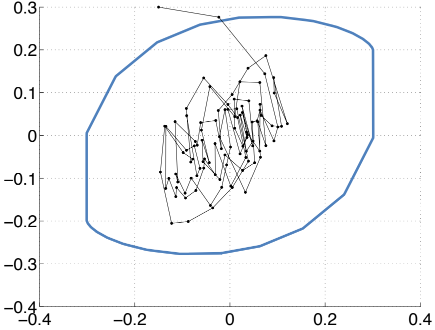
$$Q := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R := 10$$

Offline Design - Compute Minimal Invariant Set

1. Choose a stabilizing controller K so that $\|A + BK\| < 1$
2. Compute the minimal robust invariant set $\mathcal{E} = F_\infty$ for the system $x^+ = (A + BK)x + w$, $w \in \mathbb{W}$

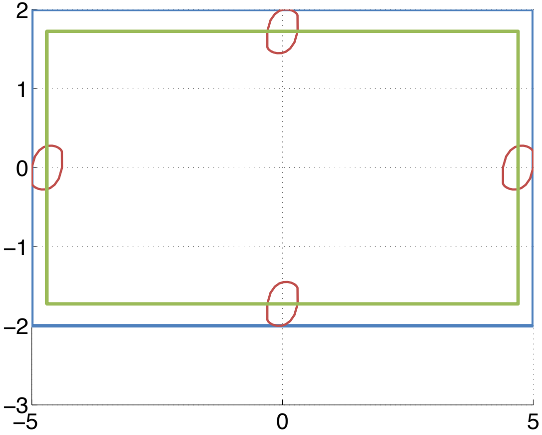
We take the LQR controller for $Q = I$, $R = 1$:

$$K := \begin{bmatrix} -0.5198 & -0.9400 \end{bmatrix}$$



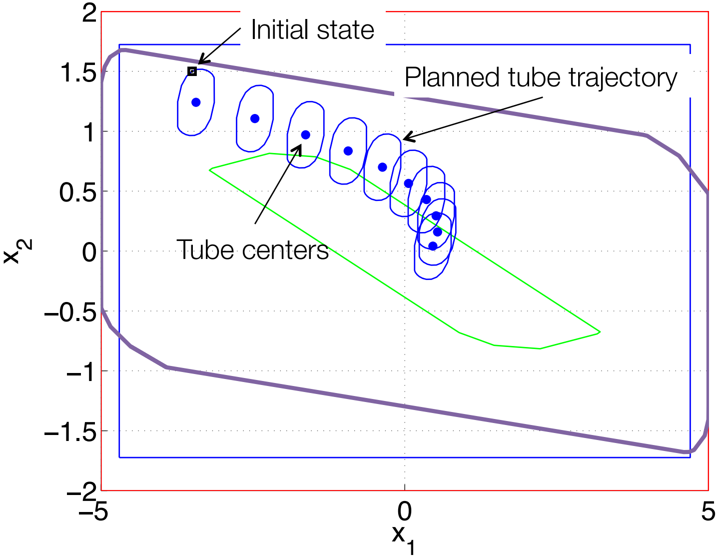
Evolution of the system $x^+ = (A + BK)x + w$ for $x_0 = \begin{bmatrix} -0.1 & 0.2 \end{bmatrix}^T$

Offline Design - Tighten State Constraints

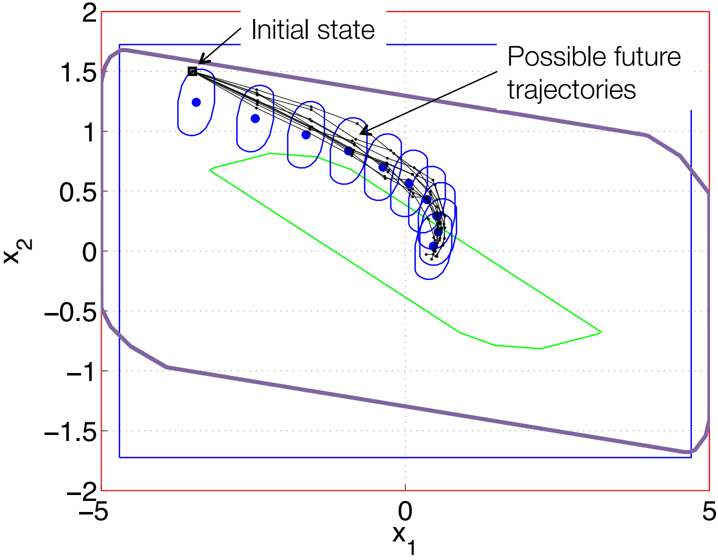


Blue : Original constraint set $\mathcal{X}$
Red : Error set $\mathcal{E}$
Green : Tightened constraints $\mathcal{X} \ominus \mathcal{E}$

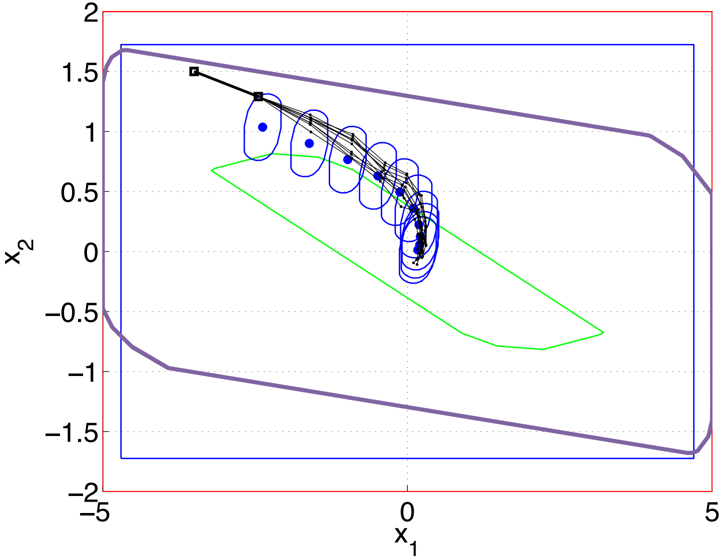
Tubes - Example



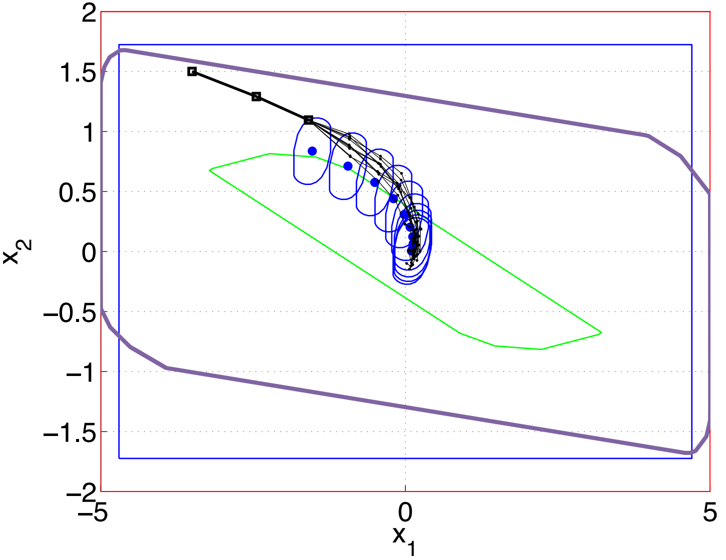
Tubes - Example



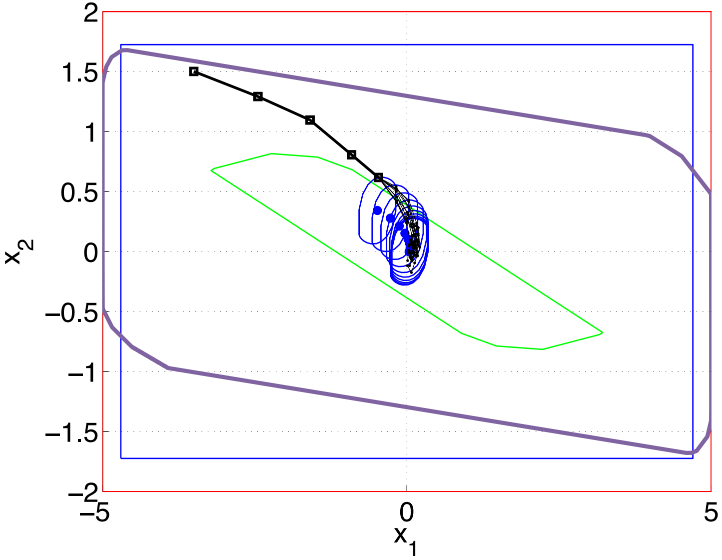
Tubes - Example



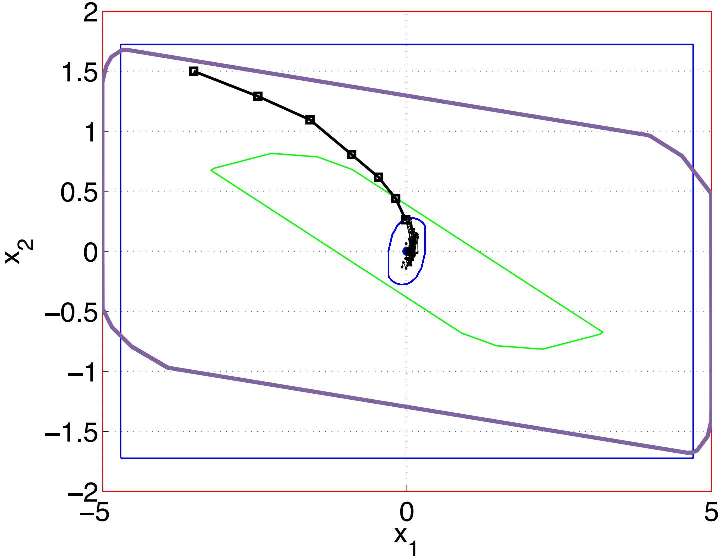
Tubes - Example



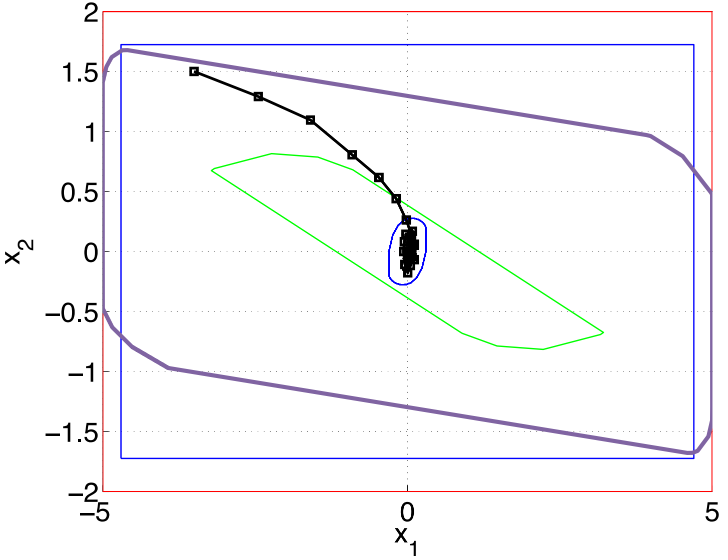
Tubes - Example



Tubes - Example



Tubes - Example



Tube MPC - Summary

Idea:

- Split input into two parts: One to steer system (v), one to compensate for the noise (Ke)

$$u = Ke + v$$

- Optimize for the nominal trajectory, ensuring that any deviations stay within constraints

Benefits:

- Less conservative than open-loop robust MPC (we're now actively compensating for noise in the prediction)
- Works for unstable systems
- Optimization problem to solve is simple

Cons:

- Sub-optimal MPC (optimal is extremely difficult)
- Reduced feasible set when compared to nominal MPC
- We need to know what $\mathbb{W}$ is (this is usually not realistic)

Robust MPC for Uncertain Systems - Summary

Idea

- Compensate for noise in prediction to ensure all constraints will be met

Cons

- Complex (some schemes are simple to implement, like tubes, but complex to understand)
- Must know the largest noise $\mathbb{W}$
- Often very conservative
- Feasible set may be small

Benefits

- Feasible set is invariant - we know exactly when the controller will work
- Easier to tune - knobs to tradeoff robustness against performance