Model Predictive Control

Lecture: Practical Model Predictive Control

Colin Jones

Laboratoire d'Automatique, EPFL

Elements of a Stabilizing and Invariant Controller

Finite-time optimal control

$$
\begin{aligned}
V_N^\star(x_0) = \min \ & \sum_{i=0}^{N-1} l(x_i, u_i) + V_f(x_N) \\
\text{s.t. } & x_{i+1} = f(x_i, u_i) \\
& (x_i, u_i) \in \mathbb{X}, \mathbb{U} \\
& x_N \in \mathcal{X}_f
\end{aligned} \tag{1}
$$

Truncate after a finite horizon:

- V_f : Approximates the 'tail' of the cost
- $\mathcal{X}_f$: Approximates the 'tail' of the constraints

Optimal control law: $\kappa_N(x) := u_0^\star$
where $u^\star := \left\{u_0^\star, \ldots, u_{N-1}^\star\right\}$ is the optimizer of (1)

What conditions do we need to place on V_f, $\mathcal{X}_f$ and l to ensure recursive feasibility and stability?

Stability of MPC - Main Result

If we can choose/find an $\mathcal{X}_f$, κ_f, V_f and l such that:

1. The stage cost is a positive definite function, i.e. it is strictly positive and only zero at the origin
2. The terminal set is **invariant** under the local control law $\kappa_f(x)$:

   $$x^+ = Ax + B\kappa_f(x) \in \mathcal{X}_f \quad \text{for all } x \in \mathcal{X}_f$$

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

   $$\mathcal{X}_f \subseteq \mathbb{X},\ \kappa_f(x) \in \mathbb{U} \text{ for all } x \in \mathcal{X}_f$$

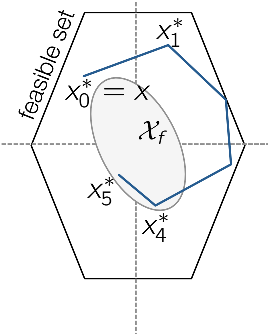
3. Terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_f$:

   $$V_f(x^+) - V_f(x) \leq -l(x, \kappa_f(x)) \text{ for all } x \in \mathcal{X}_f$$

Thm: The closed-loop system under the MPC control law $u_0^\star(x)$ is stable and the system $x^+ = Ax + Bu_0^\star(x)$ is invariant in the feasible set $\mathbb{X}_N$.

Stability of MPC - Outline of the Proof

- Assume feasibility of x and let $[u_0^\star,\ u_1^\star,\ \ldots,\ u_{N-1}^\star]$ be the optimal control sequence computed at x

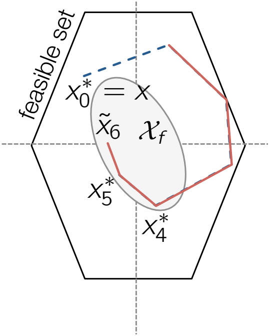


Stability of MPC - Outline of the Proof

- Assume feasibility of x and let $[u_0^\star,\ u_1^\star,\ \dots,\ u_{N-1}^\star]$ be the optimal control sequence computed at x

- At x^+, $[u_1^\star,\ u_2^\star,\ \dots,\ \kappa_f(x_N^\star)]$ is feasible:

$$x_N \text{ is in } \mathcal{X}_f \to \kappa_f(x_N^\star) \text{ is feasible}$$
$$\text{and } x_{N+1} = Ax_N^\star + B\kappa_f(x_N^\star) \text{ in } \mathcal{X}_f$$



$\Rightarrow$ **Terminal constraint provides recursive feasibility**

Stability of MPC - Outline of the Proof

$$J^*(x_0) = \sum_{i=0}^{N-1} l(x_i^*, u_i^*) + V_f(x_N^*)$$

Feasible, sub-optimal sequence for x_1 : $[u_1^\star,\ u_2^\star,\ \ldots,\ \kappa_f(x_N^\star)]$

$$\begin{aligned}
J^*(x_1) &\leq \sum_{i=1}^{N} l(x_i^*, u_i^*) + V_f(\tilde{x}_{N+1}) \\
&= \sum_{i=0}^{N-1} l(x_i^*, u_i^*) + V_f(x_N^*) - l(x_0^*, u_0^*) + V_f(\tilde{x}_{N+1}) - V_f(x_N^*) + l(x_N^*, \kappa_f(x_N^*)) \\
&= J^*(x_0) - l(x, u_0^*) + \underbrace{V_f(\tilde{x}_{N+1}) - V_f(x_N^*) + l(x_N^*, \kappa_f(x_N^*))}_{V_f(x) \text{ is a Lyapunov function: } \leq 0}
\end{aligned}$$

$J^\star(x)$ is a Lyapunov function $\rightarrow$ (Lyapunov) Stability

Choice of Terminal Sets and Functions

How do we choose V_f, $\mathcal{X}_f$, κ_f to satisfy stability conditions?

Can be difficult in general, but one case is constructive:

$$f(x,u) = Ax + Bu \qquad \mathbb{X} \text{ and } \mathbb{U} \text{ polytopes} \qquad l(x,u) = x^T Q x + u^T R u$$

Define the terminal controller as the optimal unconstrained LQR control law, and the terminal weight as the optimal LQR cost:

$$\kappa_f(x) = Kx \qquad K = -(R + B^T P B)^{-1} B^T P A$$

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

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

Choose the terminal weight to be the optimal LQR cost:

$$V_f(x) := x^T P x = \sum_{i=0}^{\infty} {x_i}^T Q x_i + {x_i}^T K^T R K x_i$$

Choose the terminal set $\mathcal{X}_f$ to be the maximum invariant set for the closed-loop system $x^+ = (A + BK)x$ subject to $\mathcal{X}_f \subset \mathbb{X}$, $K\mathcal{X}_f \subset \mathbb{U}$

Choice of Terminal Sets and Functions

1. The stage cost is a positive definite function

$$l(x,u) = x^T Q x + u^T R u > 0 \text{ for all } x, u \neq 0$$

2. The terminal set is **invariant** under the local control law $u = Kx$.
   $\mathcal{X}_f$ has been defined as the largest invariant set for the terminal control law.

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

$$\begin{aligned} V_f(x_1) - V_f(x_0) &= {x_1}^T P x_1 - {x_0}^T P x_0 \\ &= \sum_{i=1}^{\infty} {x_i}^T (Q + K^T R K) x_i - \sum_{i=0}^{\infty} {x_i}^T (Q + K^T R K) x_i \\ &= -{x_0}^T (Q + K^T R K) x_0 \\ &= -l(x_0, K x_0) \end{aligned}$$

Example: Unstable Linear System

System dynamics:

$$x^+ = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

Constraints:

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

Stage cost:

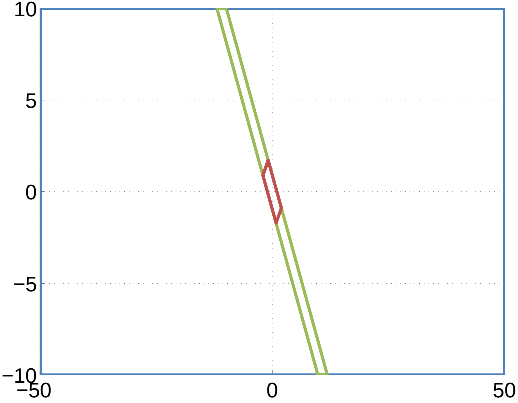
$$l(x, u) := x^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + u^T u$$

Horizon: $N = 10$

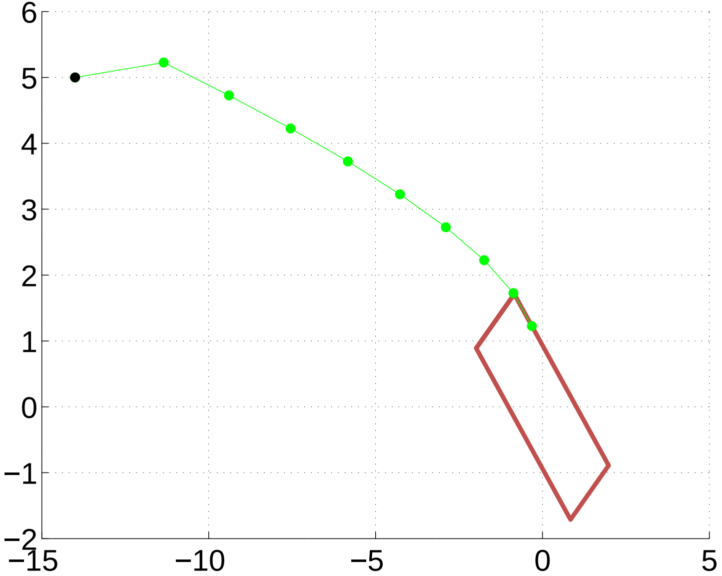
Example: Designing MPC Problem

1. Compute the optimal LQR controller and cost matrices: K, P
2. Compute the maximal invariant set $\mathcal{X}_f$ for the closed-loop linear system $x^+ = (A + BK)x$ subject to the constraints

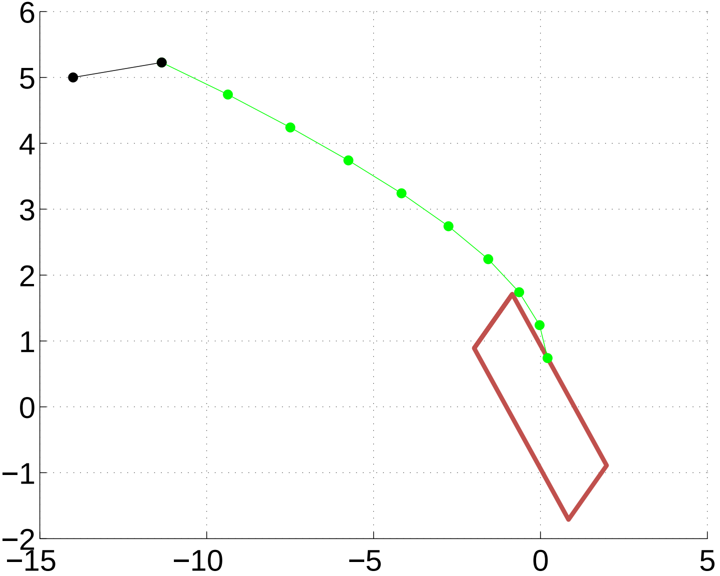
$$X_{\text{cl}} := \left\{ x \;\middle|\; \begin{bmatrix} A_x \\ A_u K \end{bmatrix} x \le \begin{bmatrix} b_x \\ b_u \end{bmatrix} \right\}$$



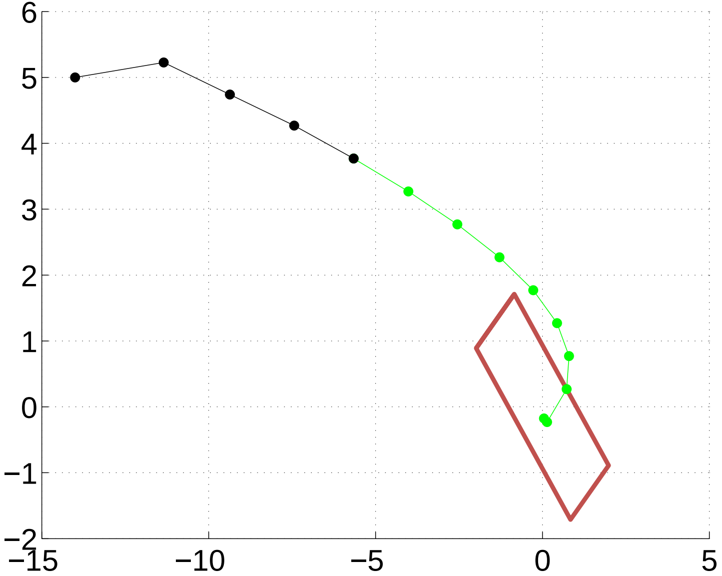
Example: Closed-loop behaviour



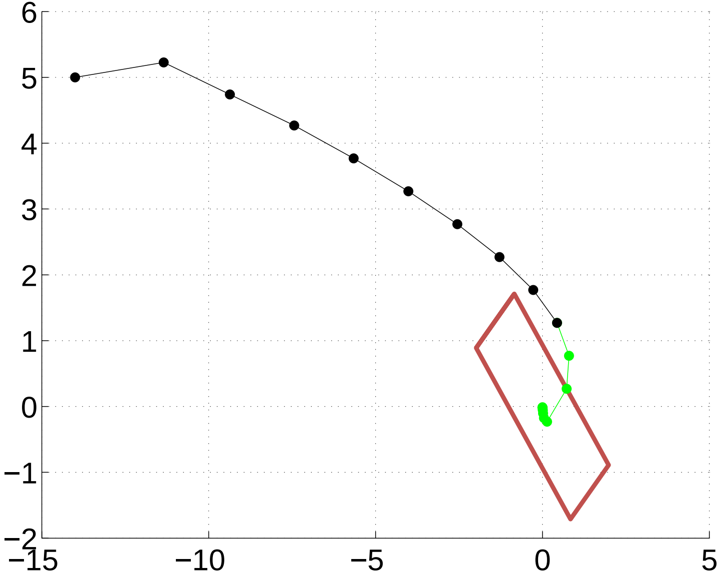
Example: Closed-loop behaviour



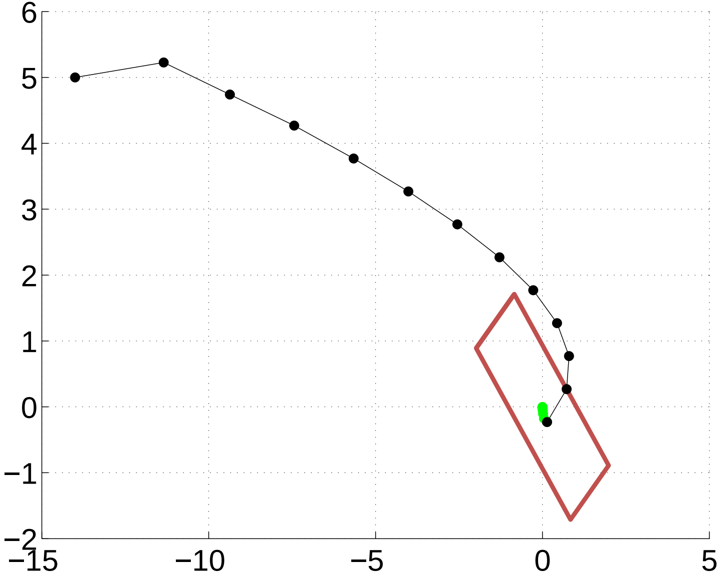
Example: Closed-loop behaviour



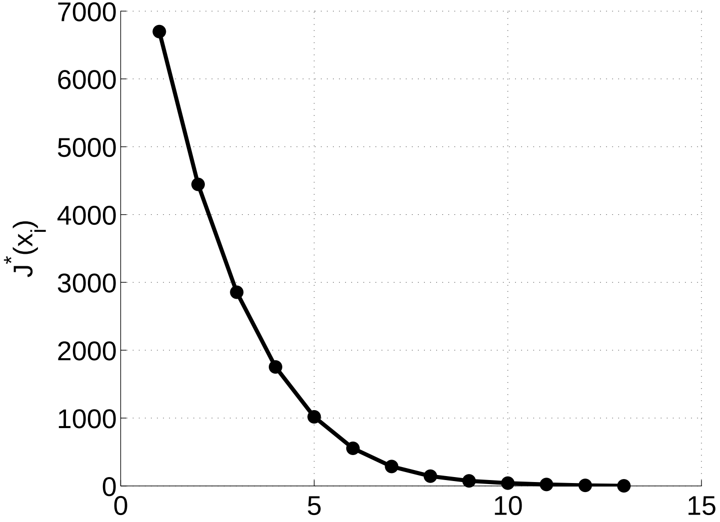
Example: Closed-loop behaviour



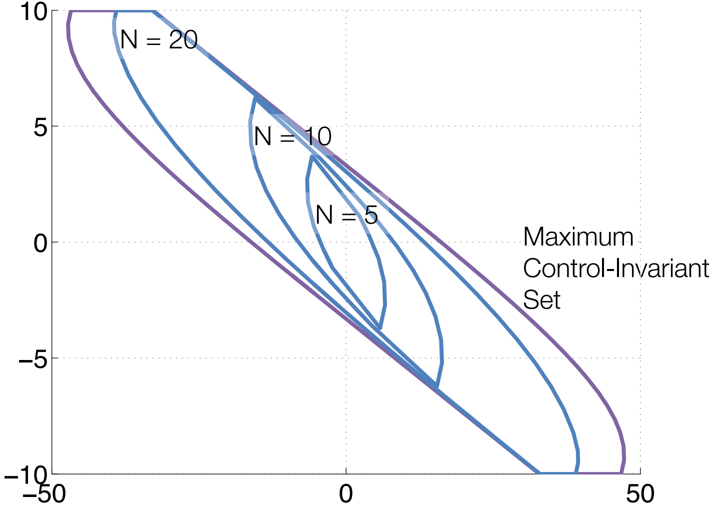
Example: Closed-loop behaviour



Example: Lyapunov Decrease of Optimal Cost



Example: Impact of Horizon



The horizon can have a strong impact on the region of attraction.

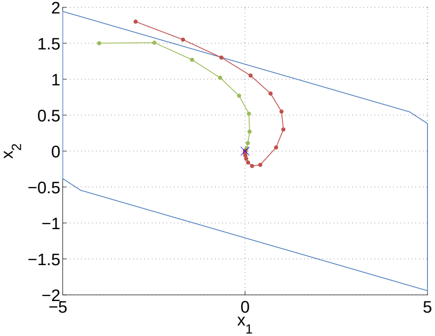
Outline

1. MPC: Practical Issues
2. Enlarging the Feasible Set
   - MPC without Terminal Set
   - Soft Constrained MPC
3. Tracking
4. Offset-free control

MPC: Practical Issues

Feasible Set

- Constraints restrict the set of states for which the optimization problem is feasible.
- MPC controller is only defined in the feasible set, where a solution exists



Example: Double Integrator

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

$$-0.5 \le u \le 0.5$$

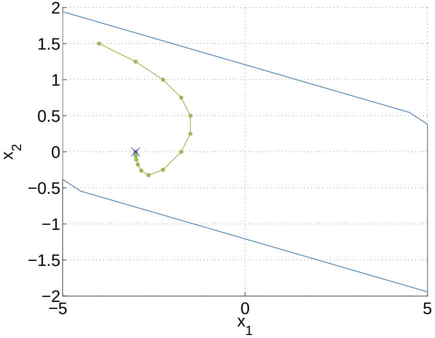
$$-5 \le x_i \le 5, \; i = 1, 2$$

→ Want the feasible set to be as large as possible

MPC: Practical Issues

Tracking

- Classic MPC problem: Regulation to the origin
- Common task in practice: Tracking of non-zero output set points



Example: Double Integrator

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

$$-0.5 \leq u \leq 0.5$$

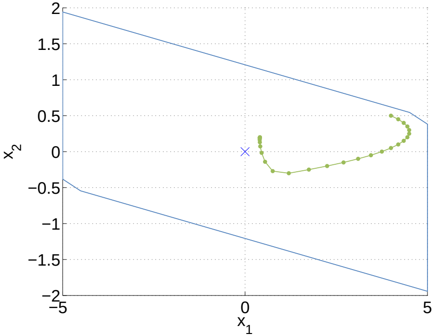
$$-5 \leq x_i \leq 5, \ i = 1, 2$$

→ Want to use MPC for tracking

MPC: Practical Issues

Disturbance rejection

- Constant disturbance causes offset from the origin / the desired set point



Example: Double Integrator

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

$$d = 0.2$$

$$-0.5 \leq u \leq 0.5$$

$$-5 \leq x_i \leq 5, \ i = 1, 2$$

→ Want to remove offset such that system converges to desired set point.

Outline

1. MPC: Practical Issues

2. Enlarging the Feasible Set
    - MPC without Terminal Set
    - Soft Constrained MPC

3. Tracking

4. Offset-free control

Review: Stability of MPC

Assume that

1. The stage cost is a positive definite function, i.e. it is strictly positive and only zero at the origin
2. The terminal set is **invariant** under the local control law $\kappa_f(x)$:

$$x^+ = Ax + B\kappa_f(x) \in \mathcal{X}_f \quad \text{for all } x \in \mathcal{X}_f$$

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

$$\mathcal{X}_f \subseteq \mathbb{X},\ \kappa_f(x) \in \mathbb{U} \text{ for all } x \in \mathcal{X}_f$$

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

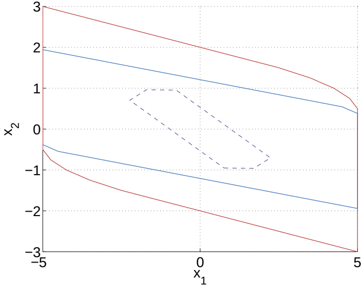
$$V_f(x^+) - V_f(x) \leq -l(x, \kappa_f(x)) \text{ for all } x \in \mathcal{X}_f$$

Thm: The closed-loop system under the MPC control law $u_0^\star(x)$ is stable and the system $x^+ = Ax + Bu_0^\star(x)$ is invariant in the feasible set $\mathbb{X}_N$.

MPC without Terminal Set

Motivation

- Terminal constraint reduces feasible set



  - Blue line: Feasible set with terminal constraint
  - Red line: Feasible set without terminal constraint
  - Dashed line: Terminal set

- Potentially adds large number of extra constraints
- Adds state constraints to problems with only input constraints

Goal: MPC without terminal constraint with guaranteed stability

Note: Feasible set without terminal constraint is not invariant.

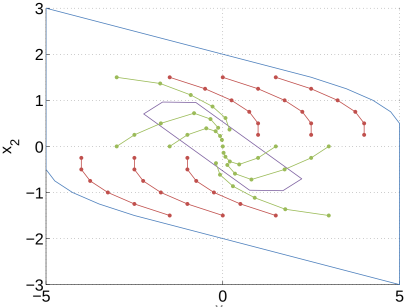
MPC without Terminal Set

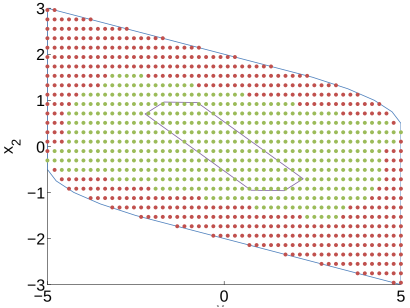
We can remove terminal constraint while maintaining stability if

- initial state lies in sufficiently small subset of feasible set
- N is sufficiently large

such that terminal state satisfies terminal constraint without enforcing it in the optimization.

$\Rightarrow$ Solution of the finite horizon MPC problem corresponds to the infinite horizon solution





MPC without Terminal Set: Discussion

Advantage: Controller defined in a larger feasible set

Disadvantage: Characterization of region of attraction or specification of required horizon length extremely difficult

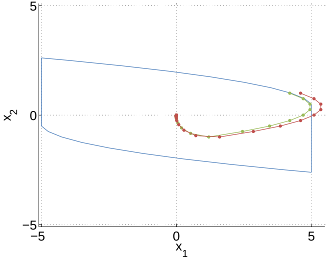
Remarks:

- Terminal constraint provides a sufficient condition for stability:
  Region of attraction without terminal constraint may be larger than for MPC with terminal constraint
- In practice: Enlarge horizon and check stability by sampling
- With larger horizon length N, region of attraction approaches maximum control invariant set

Soft constrained MPC: Concept

Motivation:

- State constraints may lead to infeasibility (also without terminal constraint)
- Controller must provide some input in every circumstance



- Input constraints often represent actuator limitations
  $\rightarrow$ generally have to be considered and satisfied
- State constraints often represent performance or comfort constraints
  $\rightarrow$ could be temporarily violated if necessary
- Soft constraints are common practice in industry

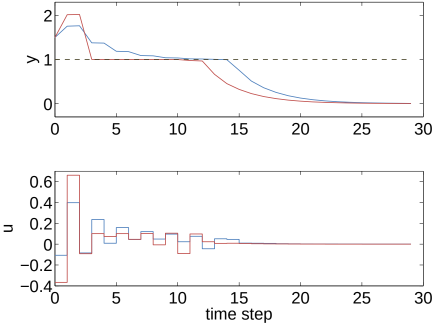
Objectives

Goals:

- Minimize the duration of the violation
- Minimize the size of the violation

These can be conflicting goals, i.e. reduction in size of violation can only be achieved at cost of large increase in duration of violation

→ Multi-objective problem

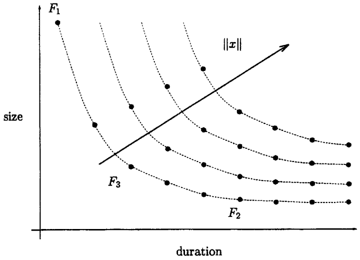


Red: Minimum time of violation

Blue: Minimum size of violation

Multi-objective Problem

For given system and horizon can plot pareto optimal size/duration curve for different initial conditions:



Best operation points lie on pareto optimal curve:

- points below cannot be attained
- points above are inferior

$\rightarrow$ Operation at pareto optimality is in general difficult and only approximately achieved

Best operation point minimum time or minimum violation?

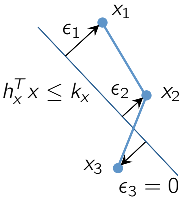
Depends on application, e.g.:

- if product must be discarded during constraint violation, goal is minimum time of violation
- if large constraint violations can lead to process shutdown or exceptions goal is minimum amount of violation

Soft constrained MPC problem setup

$$\min_{\mathbf{u}} \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + \rho(\epsilon_i) + x_N^T P x_N + \rho(\epsilon_N)$$

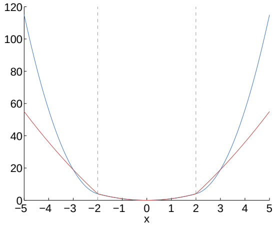
$$\begin{aligned} \text{s.t. } & x_{i+1} = Ax_i + Bu_i \\ & H_x x_i \leq k_x + \epsilon_i, \\ & H_u u_i \leq k_u, \\ & \epsilon_i \geq 0 \end{aligned}$$



- Relax state constraints by introducing so called **slack variables** $\epsilon_i \in \mathbb{R}^p$
- Penalize amount of constraint violation in the cost by means of penalty $\rho(\epsilon_i)$

How to choose penalty?

- Quadratic penalty: $\rho(\epsilon_i) = \epsilon_i^T S \epsilon_i$
- Quadratic and linear norm penalty: $\rho(\epsilon_i) = \epsilon_i^T S \epsilon_i + s\|\epsilon_i\|_{1/\infty}$



Blue: $x^T x + 10\epsilon^T \epsilon$
Red: $x^T x + 10\|\epsilon\|_1$

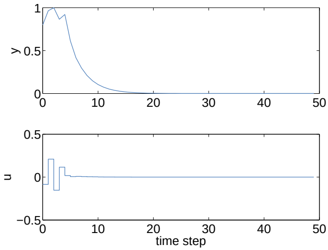
Example

Consider the third-order non-minimum phase system

$$x^+ = \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$
$$y = \begin{bmatrix} -1 & 0 & 2 \end{bmatrix}$$

- Constraint: $-1 \leq y \leq 1$
- Controller parameters: $Q = C^T C$, $R = 1$ and $N = 20$.

Initial condition $x(0) = \begin{bmatrix} 0.8 & 0.8 & 0.8 \end{bmatrix}^T$:

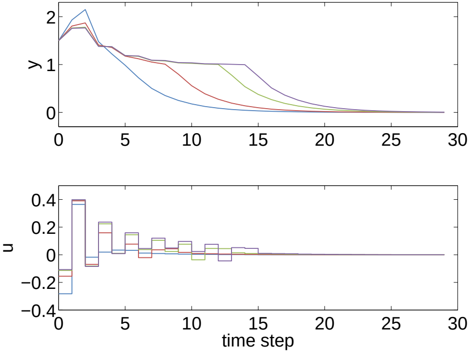


Soft constraints with quadratic penalty

Properties of the quadratic penalty:

- Well-posed quadratic program (positive definite Hessian)
- Increase in S leads to 'hardening' of the soft constraints

Example:



Initial condition:
$x(0) = \begin{bmatrix} 1.5 & 1.5 & 1.5 \end{bmatrix}^T$

Blue:	$S = 1$
Red:	$S = 10$
Green:	$S = 50$
Violet:	$S = 100$

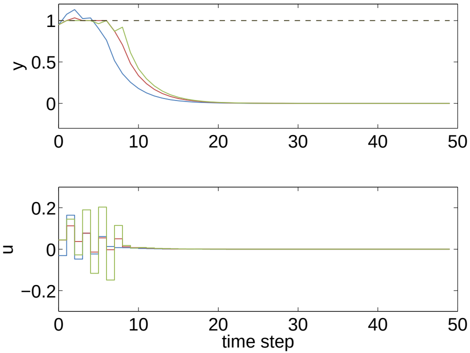
→ Increase in S leads to reduced size of violation but longer duration

Soft constraints with quadratic and linear penalty

Properties of the linear penalty:

- Allows for exact penalties: If weight s is chosen large enough, constraints are satisfied if possible

Example:



Initial condition:
$x(0) = \begin{bmatrix} .95 & .95 & .95 \end{bmatrix}^T$
Penalty: 1-norm, S= 20

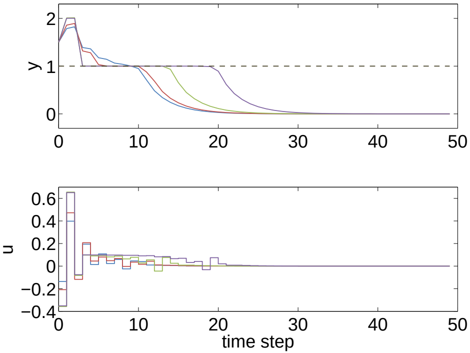
Blue: $s = 0.1$
Red: $s = 10$
Green: $s = 25$

Soft constraints with quadratic and linear penalty

Properties of the linear penalty:

- Allows for exact penalties: If weight s is chosen large enough, constraints are satisfied if possible
- Increasing s results in increasing peak violations and decreasing duration
- Large linear penalties make tuning difficult and cause numerical problems

Example:



Initial condition:
$x(0) = \begin{bmatrix} 1.5 & 1.5 & 1.5 \end{bmatrix}^T$
Penalty: 1-norm, S= 20

Blue: $s = 0.1$
Red: $s = 10$
Green: $s = 25$

Simplification: Separation of objectives

1. Minimize violation over the horizon:

$$
\begin{aligned}
\epsilon^{\min} = \text{argmin}_{\mathbf{u},\epsilon} \epsilon_i^T S \epsilon_i + s^T \epsilon_i \\
\text{s.t. } x_{i+1} &= Ax_i + Bu_i \\
H_x x_i &\leq K_x + \epsilon_i, \\
H_u u_i &\leq K_u, \\
\epsilon_i &\geq 0
\end{aligned}
$$

2. Optimize for controller performance:

$$
\min_{\mathbf{u}} \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T P x_N
$$

$$
\begin{aligned}
\text{s.t. } x_{i+1} &= Ax_i + Bu_i \\
H_x x_i &\leq k_x + \epsilon_i^{\min}, \\
H_u u_i &\leq k_u,
\end{aligned}
$$

⇒ Advantage: Simplifies tuning, constraints will be satisfied if possible
⇒ Disadvantage: Requires solution of two optimization problems

Recap: Soft constraints

- Soft constraints recover feasibility of the optimization when constraints cannot be satisfied
- Allow for a variety of violation duration vs. size tradeoffs
- Good closed-loop properties

$\Rightarrow$ Generally applied in practice

Note: standard methods for soft constrained MPC do not provide a stability guarantee for open-loop unstable systems

$\Rightarrow$ Recent developments towards soft constrained MPC with stability guarantees[1]

[1] Preliminary work in M.N.Zeilinger, C.N. Jones and M. Morari, Robust stability properties of soft constrained MPC, Conf. on Decision and Control, 2010

Outline

1. MPC: Practical Issues

2. Enlarging the Feasible Set
   - MPC without Terminal Set
   - Soft Constrained MPC

3. Tracking

4. Offset-free control

Tracking problem: Introduction

Standard MPC problem regulates system state to the origin

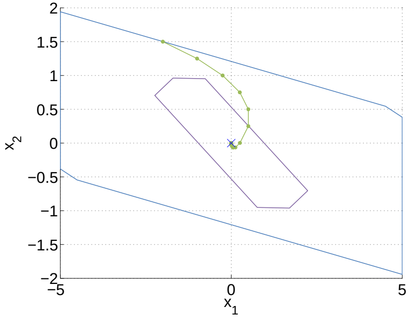
$$
\begin{aligned}
u^\star(x) := \text{argmin} \quad & x_N^T Q_f x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\
\text{s.t.} \quad & x_0 = x && \text{measurement} \\
& x_{i+1} = A x_i + B u_i && \text{system model} \\
& C x_i + D u_i \leq b && \text{constraints} \\
& R \succ 0, Q \succ 0 && \text{performance weights}
\end{aligned}
$$

Common task: Tracking of non-zero output set points

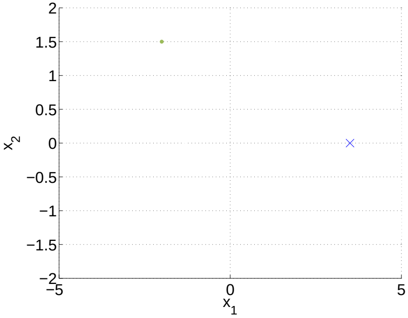
$\Rightarrow$ How can we modify the MPC problem to achieve tracking?

Tracking problem: Introduction

Regulation to origin:



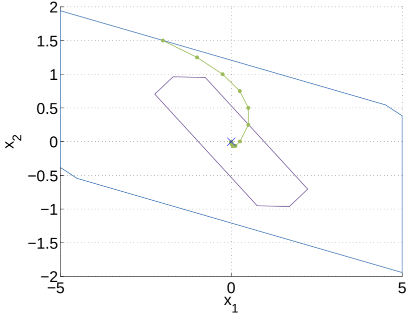
Non-zero target:



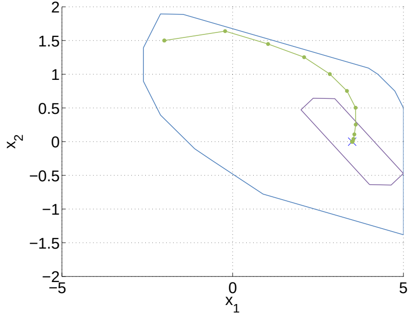
- Given output target y_r, how do we obtain a corresponding state target x_r?

Tracking problem: Introduction

Regulation to origin:



Non-zero target:



- Given output target y_r, how do we obtain a corresponding state target x_r?
- How do we adapt MPC cost to control the system to the target state?
- How do we choose terminal set and when is MPC problem feasible with respect to target?

Tracking problem

Consider the linear system model

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$.

- Assumption: State can be measured.
- Constraints: $x \in \mathbb{X}$, $u \in \mathbb{U}$ with constraint sets

$$\mathbb{X} = \{x \mid H_x x \leq k_x\}, \mathbb{U} = \{x \mid H_u u \leq k_u\}$$

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

Note: Using this framework we can track sequence of constant targets, i.e. a piecewise constant reference. We will not cover time-varying references in this lecture.

Target state corresponding to output reference

- The reference is achieved by the target state x_s if $y_s = Cx_s = r$
- Target state should be a steady-state, such that there exists an input that keeps system at target, i.e. $x_s = Ax_s + Bu_s$

$\Rightarrow$ Target condition:

$$\begin{aligned} x_s &= Ax_s + Bu_s \\ Cx_s &= r \end{aligned} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix}}_{(n_x+n_y)\times(n_x+n_u)} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Steady-state target problem

- In the presence of constraints: (x_s, u_s) has to satisfy state and input constraints.
- Compute steady-state (x_s, u_s) corresponding to reference r:

$$\begin{aligned} \min \ & u_s^T R_s u_s \\ \text{s.t.} \ & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ & H_x x_s \leq k_x \\ & H_u u_s \leq k_u \end{aligned}$$

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

$$\begin{aligned} \min \ & (Cx_s - r)^T Q_s (Cx_s - r) \\ \text{s.t.} \ & x_s = Ax_s + Bu_s \\ & H_x x_s \leq k_x \\ & H_u u_s \leq k_u \end{aligned}$$

Delta-Formulation for tracking

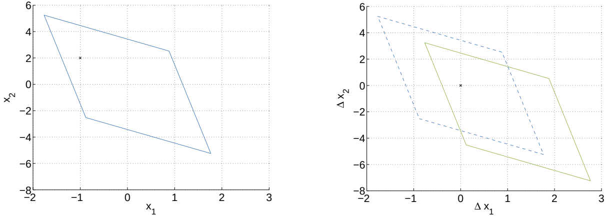
Idea: Treat set point tracking as regulation problem with a coordinate transformation

- Define deviation variables that (in the linear case) satisfy the same model equations:

$$\begin{aligned} \Delta x &= x - x_s \\ \Delta u &= u - u_s \end{aligned} \quad \Rightarrow \quad \begin{aligned} \Delta x_{k+1} &= x_{k+1} - x_s \\ &= Ax_k + Bu_k - (Ax_s + Bu_s) \\ &= A\Delta x_k + B\Delta u_k \end{aligned}$$

- Constraints for deviation variables:

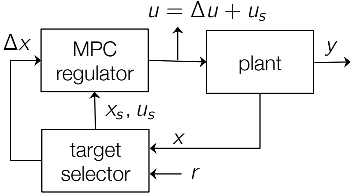
$$\begin{aligned} H_x x \le k_x \Rightarrow & \quad H_x \Delta x \le k_x - H_x x_s \\ H_u \le k_u \Rightarrow & \quad H_u \Delta u \le k_u - H_u u_s \end{aligned}$$



MPC problem for tracking

- Obtain target steady-state corresponding to reference r. [2]
- Initial state $\Delta x = x - x_s$.
- Apply regulation problem to new system in Delta-Formulation:

$$\begin{aligned} \min \ & \sum_{i=0}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + V_f(\Delta x_N) \\ \text{s.t. } & \Delta x_0 = \Delta x \\ & \Delta x_{i+1} = A \Delta x_i + B \Delta u_i \\ & H_x \Delta x_i \leq k_x - H_x x_s \\ & H_u \Delta u_i \leq k_u - H_u u_s \\ & \Delta x_N \in \mathcal{X}_f \end{aligned}$$



- Find optimal sequence of $\mathbf{\Delta u}^*$
- Input applied to the system is $u_0^* = \Delta u_0^* + u_s$

[2] If the target steady-state is uniquely defined by the reference, we can also include the target condition as a constraint in the MPC problem.

MPC problem for tracking

Convergence

Assume target is feasible with $x_s \in \mathbb{X}, u_s \in \mathbb{U}$ and choose terminal weight $V_f(x)$ and constraint $\mathcal{X}_f$ as in the regulation case satisfying:

- $\mathcal{X}_f \subseteq \mathbb{X}$, $Kx \in \mathbb{U}$ for all $x \in \mathcal{X}_f$
- $V_f(x^+) - V_f(x) \leq -l(x, Kx)$ for all $x \in \mathcal{X}_f$

If in addition the target reference x_s, u_s is such that

- $x_s \oplus \mathcal{X}_f \subseteq \mathbb{X}$, $K\Delta x + u_s \in \mathbb{U}$ for all $\Delta x \in \mathcal{X}_f$

then the closed-loop system converges to the target reference,
i.e. $x_k \to x_s$ and therefore $y_k = Cx_k \to r$ for $k \to \infty$

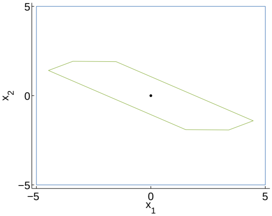
Proof: Choose local control law $\Delta u = K\Delta x$

- Invariance under local control law is directly inherited from regulation case
- Constraint satisfaction is provided by extra conditions:
  - $x_s \oplus \mathcal{X}_f \subseteq \mathbb{X} \ \rightarrow \ x \in \mathbb{X} \ \forall \Delta x = x - x_s \in \mathcal{X}_f$
  - $K\Delta x + u_s \in \mathbb{U} \ \forall \Delta x \in \mathcal{X}_f \ \rightarrow \ u \in \mathbb{U}$
- From asymptotic stability of the regulation problem: $\Delta x_k \to 0$ for $k \to \infty$

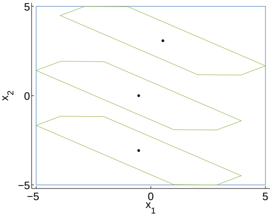
MPC for tracking: Terminal Set

For the following consideration, consider only state constraints.

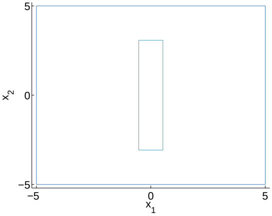
Regulation case:



Tracking using a shifted terminal set:



Set of feasible targets:

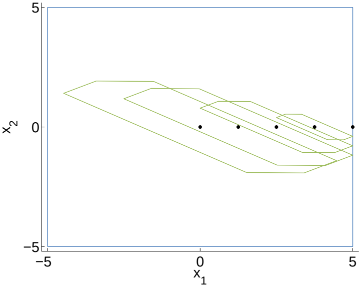


- Blue: State constraints
- Green: Terminal set

⇒ Set of feasible targets may be significantly reduced

MPC for tracking: Terminal Set

Enlarge set of feasible targets by scaling terminal set for regulation



- Scale terminal set by scaling factor α, i.e. $\mathcal{X}_f^{\text{scaled}} = \alpha\mathcal{X}_f$
- Invariance is maintained: If $\mathcal{X}_f$ is invariant, then also $\alpha\mathcal{X}_f$
- Choose scaling factor α such that state and input constraints are still satisfied

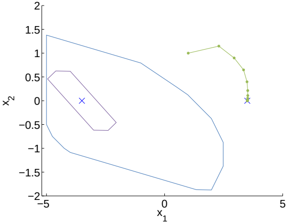
$\rightarrow$ Scaling is dependent on target

$\rightarrow$ All targets $x_s, u_s \in \mathbb{X} \times \mathbb{U}$ are feasible for constraints
Note: steady-state condition still limits the set of admissible targets

$\rightarrow$ For targets at the boundary of the constraints: $x_N = x_s$, which corresponds to a zero terminal set in the regulation case

Recap: MPC for tracking

- Set point tracking problem corresponds to regulation problem after a coordinate transformation
- If the closed-loop system is stable, set point is achieved
- Stability can be guaranteed using the same tools as for regulation
- Difficulties:
  - Terminal set is a function of the target
  - Reference change can render the optimization problem infeasible



$\rightarrow$ Recent approach for tracking resolves this issue [3]

[3] *[MPC for tracking piecewise constant references for constrained linear systems, Limon et al., Automatica 2010]*

Outline

1. MPC: Practical Issues

2. Enlarging the Feasible Set
   - MPC without Terminal Set
   - Soft Constrained MPC

3. Tracking

4. Offset-free control

Constant disturbances

Constant disturbance is acting on the system causing system trajectory to deviate from nominal dynamics.

Objective: If system is stabilized in the presence of the disturbance then it converges to set point with zero offset.

Recall:

- In classic unconstrained control introduction of an integrating mode to remove offset
- Input constraints and actuator limitations require anti-windup techniques

Approach in constrained control:

- Model the disturbance
- Use the output measurements and model to estimate the state and the disturbance
- Find control inputs that use the disturbance estimate to remove offset

Augmented model

Incorporate disturbance model assuming integral disturbance dynamics

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

with $d \in \mathbb{R}^{n_d}$.

Only restriction on choice of B_d, C_d: observability of the augmented model

The augmented system is observable if and only if (A, C) is observable and

$$\begin{bmatrix} A - I & B_d \\ C & C_d \end{bmatrix} \text{ has full column rank, i.e. } \operatorname{rank} = n_x + n_d$$

$\Rightarrow$ Maximal dimension of the disturbance: $n_d \leq n_y$

Intuition: At steady-state $\begin{bmatrix} A - I & B_d \\ C & C_d \end{bmatrix} \begin{bmatrix} x_s \\ d_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s \end{bmatrix}$ and given y_s, d_s must be uniquely defined.

Linear state estimation

Design state and disturbance estimator based on the augmented model:

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (C\hat{x}_k + C_d\hat{d}_k - y_k)$$

where $\hat{x}$, $\hat{d}$ are estimates of the state and disturbance.

Error dynamics:

$$\begin{aligned} \begin{bmatrix} x_{k+1} - \hat{x}_{k+1} \\ d_{k+1} - \hat{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\ &\quad - \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} u_k - \begin{bmatrix} L_x \\ L_d \end{bmatrix} (C\hat{x}_k + C_d\hat{d}_k - Cx_k - C_d d_k) \\ &= \left(\begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix} \right) \begin{bmatrix} x_k - \hat{x}_k \\ d_k - \hat{d}_k \end{bmatrix} \end{aligned}$$

$\Rightarrow$ Choose $L = \begin{bmatrix} L_x \\ L_d \end{bmatrix}$ such that the error dynamics are stable and converge to zero, i.e. the estimator is stable.

Additional disturbances/noise in offset-free control

Consider the augmented model that is subject to noise

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + w$$
$$y_k = Cx_k + C_d d_k + v$$

where w, v is white noise, with covariance matrices Q_e, R_e respectively. (Co-variances may be treated as design parameters or found from input-output measurements)

⇒ Use Kalman filter to estimate both the state and the integrating disturbance[4]

⇒ Optimal estimator gain L that minimizes the variance of the estimation error

---

[4] ref

Offset-free tracking

Goal: Track constant reference r, i.e. $y_k = Cx_k \to r$ for $k \to \infty$.

- New condition at steady-state:

$$
\begin{aligned}
x_s &= Ax_s + Bu_s + B_d d_s \\
y_s &= Cx_s + C_d d_s = r
\end{aligned}
$$

  - The system steady state is modified to account for effect of disturbance on state evolution
  - Target is modified to account for effect of disturbance on tracked variables

- Best forecast for steady-state disturbance is current estimate $d_s = \hat{d}$
- Adapt target condition accordingly to account for disturbance:

$$
\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B_d \hat{d} \\ r - C_d \hat{d} \end{bmatrix}
$$

Note: Same procedure for the regulation case with $r = 0$

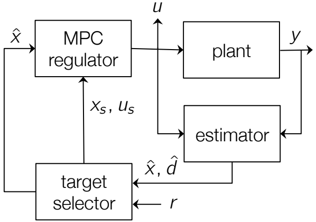
Offset-free tracking

At each sampling time

1. Estimate state and disturbance $\hat{x}$, $\hat{d}$
2. Obtain (x_s, u_s) from steady-state target problem using disturbance estimate
3. Solve MPC problem for tracking using disturbance estimate $\hat{d}$:

$$\min \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - x_s) + (u_i - u_s)^T R (u_i - u_s) + V_f(x_N - x_s)$$

$$\begin{aligned} \text{s.t. } & x_0 = \hat{x} \\ & d_i = \hat{d} \\ & x_{i+1} = Ax_i + Bu_i + d_i \\ & H_x x_i \le k_x \\ & H_u u_i \le k_u \\ & x_N - x_s \in \mathcal{X}_f \end{aligned}$$



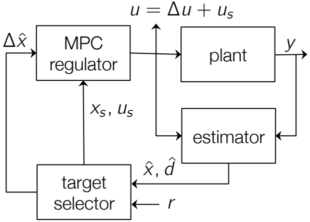
Offset-free tracking: Delta-Formulation

At each sampling time

1. Estimate state and disturbance $\hat{x}$, $\hat{d}$
2. Obtain (x_s, u_s) from steady-state target problem using disturbance estimate
3. Initial state $\Delta\hat{x} = \hat{x} - x_s$
4. Solve MPC problem for tracking in Delta-Formulation:

$$\min \sum_{i=0}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + V_f(\Delta x_N)$$

$$\begin{aligned} \text{s.t. } & \Delta x_0 = \Delta\hat{x} \\ & \Delta x_{i+1} = A\Delta x_i + B\Delta u_i \\ & H_x \Delta x_i \leq k_x - H_x x_s \\ & H_u \Delta u_i \leq k_u - H_u u_s \\ & \Delta x_N \in \mathcal{X}_f \end{aligned}$$



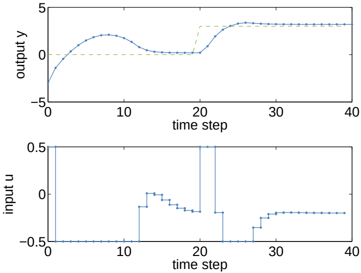
Offset-free control: Main result

- Consider the case $n_d = n_y$, i.e. number of disturbance states equal to number of measured outputs.
- Assume target steady-state problem is feasible and constraints are not active at steady-state.

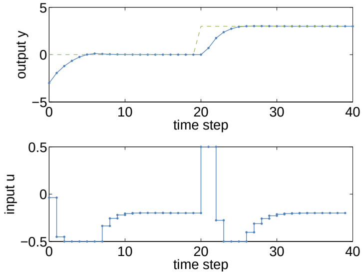
If closed-loop system converges to $\hat{x}_s, \hat{d}_s, y_s$, i.e. $\hat{x}_k \rightarrow \hat{x}_s$, $\hat{d}_k \rightarrow \hat{d}_s$, $y_k \rightarrow y_s$ as $k \rightarrow \infty$, then

$$y_k = Cx_k \rightarrow r \text{ for } k \rightarrow \infty$$

Standard tracking



Offset-free tracking



Offset-free control: Example

Double Integrator: first set point: $r = 0$, second set point: $r = 3$
1 output, 1 input, 1 modeled disturbance

Standard tracking:
Model:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$
$$-0.5 \leq u \leq 0.5$$
$$-5 \leq y \leq 5$$

Target condition:

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Offset-free tracking:
Model:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$
$$-0.5 \leq u \leq 0.5$$
$$-5 \leq y \leq 5$$

$\begin{bmatrix} I - A & -B_d \\ C & 0 \end{bmatrix}$ has full column rank

Target condition:

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -B_d \hat{d} \\ r - C_d \hat{d} \end{bmatrix}$$

Recap: Offset-free control

Approach for offset-free constrained control:

- Model the disturbance e.g. using integrating disturbance dynamics
- Use the output measurements and model to estimate the state and the disturbance (e.g. Kalman filter)
- Find control inputs that use the disturbance estimate to remove offset:
  - Modify system steady state to account for effect of disturbance
  - Modify target to account for effect of disturbance on tracked variables
  - Solve MPC problem for regulation/tracking using the disturbance estimate

Main result: If

- number of disturbance states is equal to number of outputs,
- the target steady-state problem is feasible and no constants are active at steady-state
- the closed-loop system converges,

then the target is achieved without offset.

Putting it all together

- In general state cannot be measured:
  - Use Kalman filter to estimate the state
- Design tracking problem:
  - Rewrite problem in Delta-Formulation
  - Setup target steady-state problem
  - Calculate terminal weight and scale terminal constraint to guarantee convergence
- Extend to offset-free tracking:
  - Augment model including a disturbance model
  - Augment the estimator to estimate the state and the disturbance
  - Adapt target steady-state problem using the disturbance estimate
- Possibly: Remove terminal constraint while choosing long horizon
- Introduce soft constraints to ensure feasibility at all times:
  - Introduce slack variables for constraint relaxation
  - Choose penalty on slack variables (quadratic, linear)

Exercise

Task: Implement stabilizing and invariant MPC for a simple 2-state system.

You will do this twice:

1. Manually: Compute appropriate matrices so that the problem can be solved by a standard quadratic programming solver
2. Automatically: Use the tool YALMIP to build the problem data

You will use YALMIP throughout the rest of the course, and should generally always use a tool of this sort to prevent manual translation errors.