Lecture 10: Advanced MPC and reference tracking

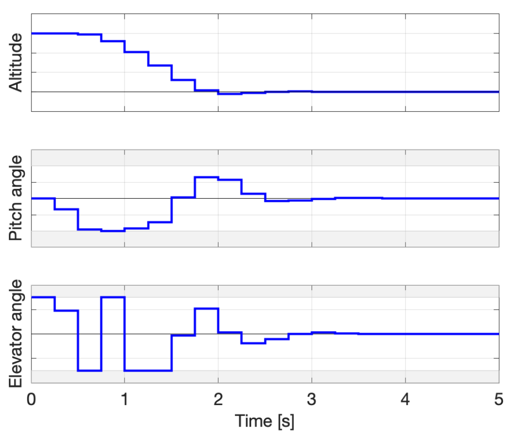
Mikael Johansson
KTH - Royal Institute of Technology

Outline

- Softening of constraints
- Terminal sets via explicit constraint checking
- Pre-stabilized predictions
- Reference-tracking via pseudo-reference

Softening of constraints: motivation

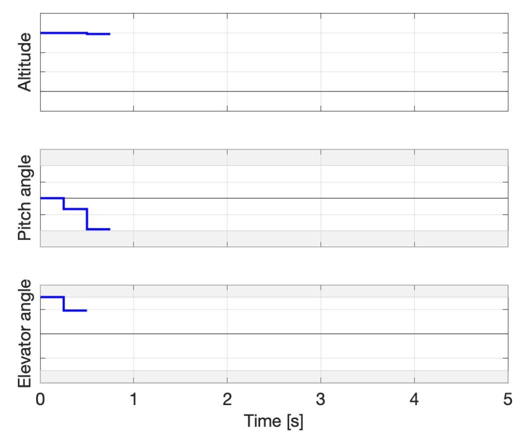
Cessna citation aircraft, 30 meter descent without rate constraints on u



Works fine, but has a slight undershoot.

Softening of constraints: motivation

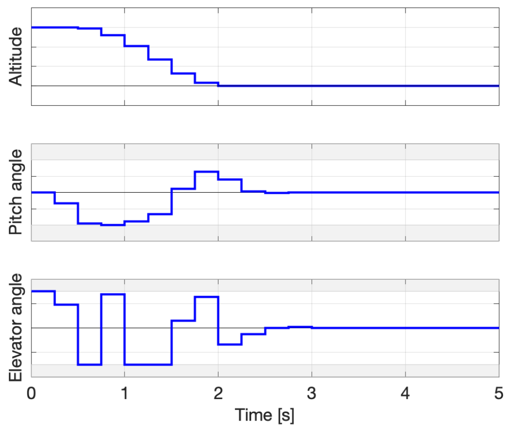
Adding constraint that altitude should be non-negative makes solver fail



(we have used terminal sets and costs; failure is due to numerics)

Softening of constraints: motivation

Softening altitude undershoot constraint avoids numerical issues.



Actual controller does not undershoot.

Softening of constraints: main idea

Idea: ensure feasibility by allowing temporary violations of constraints.

Realized using slack variables $s_k \geq 0$. Replace hard constraints

$$Hx_k \leq h$$

by

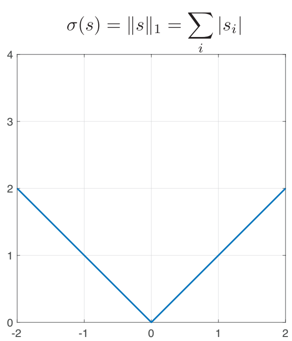
$$Hx_k \leq h + s_k$$

which can always be satisfied (possibly with large $s = (s_0, \dots, s_N)$).

Augment objective function with penalty which discourages $s \neq 0$.

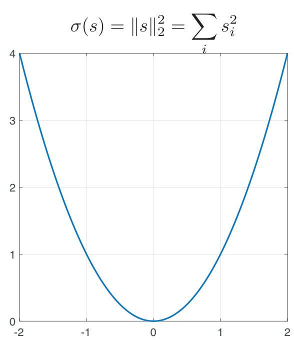
$$J = \sum_{k=1}^{N-1} q(x_{t+k|t}, u_{t+k|t}) + q_f(x_{t+N|t}) + \kappa\sigma(s)$$

Slack penalty: two-norm or one-norm?



Linear penalty:

- avoids violations if possible (large for small violations),
- can allow for large violations (small for larger violations)



Quadratic penalty:

- allows for small violations (small for small violations)
- discourages large violations (large for large violations)

Slack-penalty: two-norm or one-norm?

Example. Consider the nomial problem (left) and softened version (right)

minimize	x^2	minimize	$x^2 + \kappa\sigma(s)$
subject to	$x \leq -1$	subject to	$x \leq -1 + s$

Nominal problem has $f^\star = 1$ for $x^\star = -1$.

If we allow ϵ-constraint violation, softened problem with $\sigma(s) = s^2$ has

$$f^\star(\epsilon) = (1-\epsilon)^2 + \kappa\epsilon^2$$

so optimal violation is $\epsilon = 1/(1+\kappa)$.

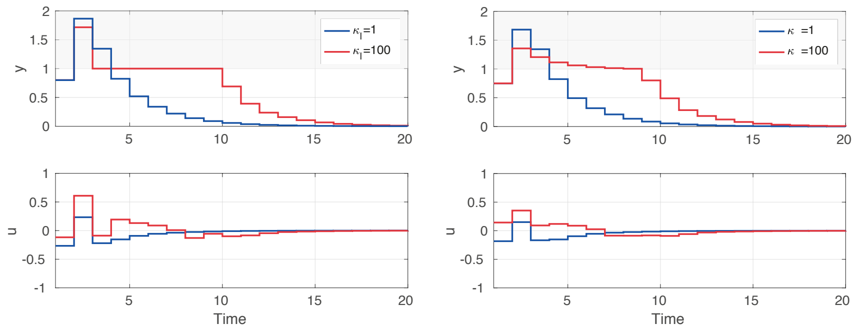
If we instead use $\sigma = |s|$, then softened problem has

$$f^\star(\epsilon) = (1-\epsilon)^2 + 2\kappa\epsilon = 1 + \epsilon^2 + 2(\kappa - 1)\epsilon$$

so if $\kappa > 1$, it will be optimal not to violate the constraint ($\epsilon = 0$)

Softening of constraints

Example. Linear system with $|u_t| \leq 1$ and $|y_t| \leq 1$. MPC uses $N = 20$.



Larger penalty $\Rightarrow$ smaller violations.
– linear penalty allows large violations of short duration
– quadratic penalty yields smaller magnitude violations for longer time

Outline

- Softening of constraints
- Terminal sets via explicit constraint checking
- Reference-tracking via pseudo-reference

Dual mode: explicit constraint checking beyond horizon

Control invariant terminal set X_f complex to calculate.
– easier to consider invariant set of closed-loop system with $u_t = -Lx_t$.

Assume that $X \cap \{x \mid -Lx \in U\}$ is on the form $\{x \mid Hx \leq h\}$. Then

$$X_f = \{x \mid Hx \leq h \ \wedge \ H(A-BL)x \leq h \ \wedge \ H(A-BL)^2x \leq h \ \wedge \ \dots\} =$$
$$= \{x_N \mid Hx_N \leq h \ \wedge \ Hx_{N+1} \leq h \ \wedge \ \dots\}$$

is an exact description of the maximal invariant set of $(A - BL)$.

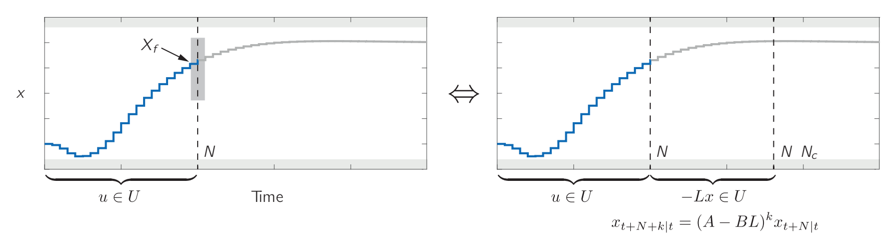
Under certain conditions (Theorem 2.3.3), X_f is finitely generated.
– allows to claim that recursive construction of X_f terminates,
– means that X_f can be represented by a finite number of inequalities
– means that $x_N \in X_f \Leftrightarrow x_{N+k} \in X \cap \{x \mid Lx \in U\}$ for all $k = 1, \dots, N_c$.

Dual mode: explicit constraint checking

Can therefore replace $x_{t+N|t} \in X_f$ by

$$x_{t+N+k|t} \in X, \qquad -Lx_{t+N+k|t} \in U$$

for $k = 0, 1, \dots, N_c$ (with N_c finite) where $x_{t+N+k|t} = (A - BL)^k x_{t+N|t}$.



Sometimes called dual mode MPC.
- mode 1: free control over control horizon N
- mode 2: fixed controll over remaining N_c steps of prediction horizon

Dual mode: explicit constraint checking beyond horizon

Invariant set computation off-line replaced by constraint checking on-line.

– advantageous when constraint sets are allowed to change on-line.

– in principle, N_c should be equal to determinedness index of invariant set

Outline

- Softening of constraints
- Terminal sets via explicit constraint checking
- Reference-tracking via pseudo-reference

The servo problem

Making the output sequence $\{y_t\}$ of the linear system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned}$$

track a given reference signal $\{r_t\}$.

From earlier, can be solved via feed-forward or feed-back (integral action)

Today, we will see how feed-forward can be incorporated in MPC

The servo problem: a state-space perspective

We have offset-free tracking if (x_t, u_t) converges to $(x^{\text{eq}}, u^{\text{eq}})$ such that

$$\begin{aligned} x^{\text{eq}} &= Ax^{\text{eq}} + Bu^{\text{eq}} \\ r &= Cx^{\text{eq}} + Du^{\text{eq}} \end{aligned}$$

If $(x^{\text{eq}}, u^{\text{eq}})$ not unique, natural to find minimum-cost stationary control:

$$\begin{array}{ll} \underset{x^{\text{eq}}, u^{\text{eq}}}{\text{minimize}} & (x^{\text{eq}})^T Q_1 x^{\text{eq}} + (u^{\text{eq}})^T Q_2 u^{\text{eq}} \\ \text{subject to} & \begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} x^{\text{eq}} \\ u^{\text{eq}} \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \end{array}$$

What can we say in the presence of constraints?

Servo-problem in presence of constraints

To guard against $(x^{\text{eq}}, u^{\text{eq}})$ being infeasible, it is natural to consider

$$\begin{array}{ll} \underset{x^{\text{eq}}, u^{\text{eq}}, \hat{r}}{\text{minimize}} & \varphi(s) \\ \text{subject to} & \begin{bmatrix} A-I & B \\ C & D \end{bmatrix} \begin{bmatrix} x^{\text{eq}} \\ u^{\text{eq}} \end{bmatrix} = \begin{bmatrix} 0 \\ r+s \end{bmatrix} \\ & x^{\text{eq}} \in X \\ & u^{\text{eq}} \in U \end{array}$$

where slack vector s is free (not necessarily positive).

Finds closest feasible set-point $\hat{r} = r + s$, tracks this value instead.

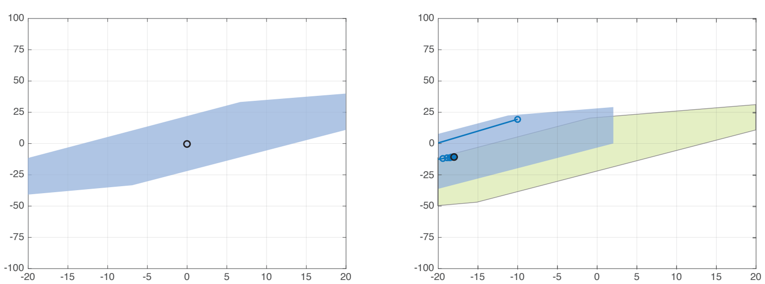
Tracking constant references with MPC

Shift coordinates to penalize *deviations* from $(x^{\text{eq}}, u^{\text{eq}})$ instead:

$$\begin{array}{ll} \text{minimize} & \sum_{k=0}^{N-1} q(x_k - x^{\text{eq}}, u_k - u^{\text{eq}}) + q_f(x_N - x^{\text{eq}}) \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k \\ & x_k \in X \\ & u_k \in U \\ & x_N - x^{\text{eq}} \in X_f \end{array}$$

Shifting terminal set can be problematic, since X_f needs to be invariant.

A problem: shifted terminal set not necessarily invariant



Maximum invariant set for $x_{\text{ref}} = 0$ (left).

Shifted terminal set (right) no longer contained in X, no longer invariant.

Also, shifted set $x_{\text{ref}} + X_f \cap X$ small, with small region of attraction.

Dealing with the terminal set

Several ways of dealing with the terminal set

1. Re-compute invariant set on-line for every new set-point
2. Use invariant set which is easy to shift (e.g. $X_f = 0$)
3. Rely on explicit constraint checking and long enough horizons

$\vdots$

For simplicity, we will use the explicit constraint checking approach.

Pseudo-reference MPC

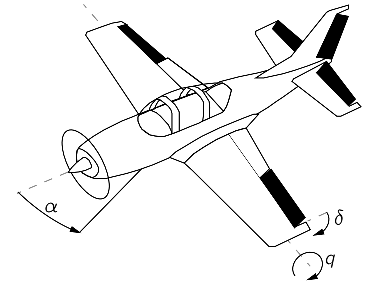
Introducing time-varying reference slacks increases flexibility further

$$
\begin{array}{ll}
\text{minimize} & \sum_{k=0}^{N-1} q(x_k - \hat{x}_k, u_k - \hat{u}_k) + \varphi(s_k) + q_f(x_N - x_r) \\
\text{subject to} & x_{k+1} = Ax_k + Bu_k \\
& x_k \in X \\
& u_k \in U \\
& x_N - \hat{x}_N \in X_f \\
& \begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ r_k + s_k \end{bmatrix}
\end{array}
$$

Increases computational cost (larger optimization problem).

Example

Control elevator surface deflection δ_k to track reference angle α_r.



$$
\begin{bmatrix} \alpha_{k+1} \\ q_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9719 & 0.0155 \\ 0.2097 & 0.9705 \end{bmatrix} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} + \begin{bmatrix} 0.0071 \\ 0.3263 \end{bmatrix} \delta_k
$$

$$
X = \{(\alpha, q) \;:\; -15 \leq \alpha \leq 30 \wedge -100 \leq q \leq 100\}
$$

$$
U = \{\delta \;:\; -25 \leq \delta \leq 25\}
$$

Example

Consider the cost criterion given by

$$
Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}. \quad Q_2 = 1
$$

The stationary optimal control is

$$
u_t = -L \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} + l_r \alpha_r = -\begin{bmatrix} 0.6854 & 1.8552 \end{bmatrix} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} + 3.9797\alpha_r
$$

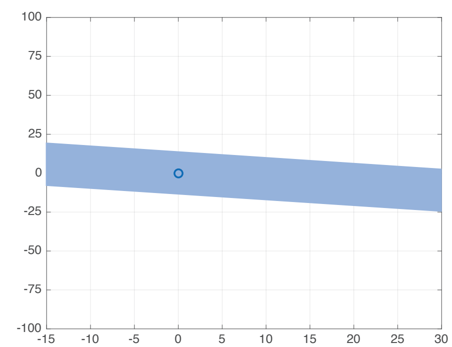
with corresponding closed-loop eigenvalues $\lambda_1 = 0.94$ and $\lambda_2 = 0.71$.

Converges to within 10% of initial error in $\log(0.1)/\log(0.94) \approx 38$ steps.

– we will use prediction horizon of $N = 40$.

Terminal sets

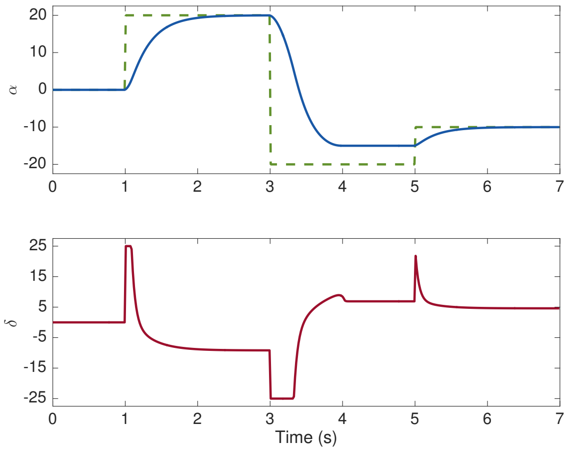
System dynamics well-damped, gives simple terminal sets.



Will use constraint checking horizon of $N_c = 10$.

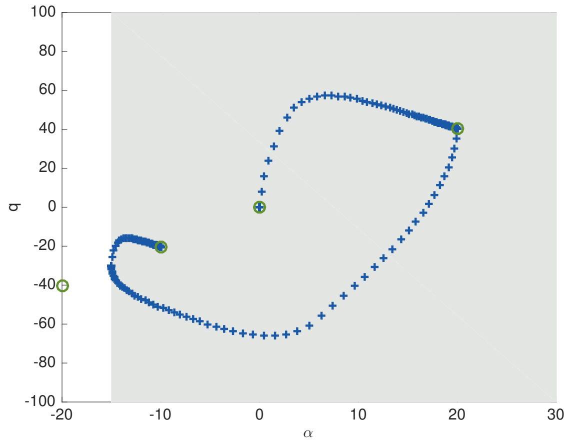
Example

Tracking of feasible and infeasible reference values ($\alpha_{\min} = -15$)



Example

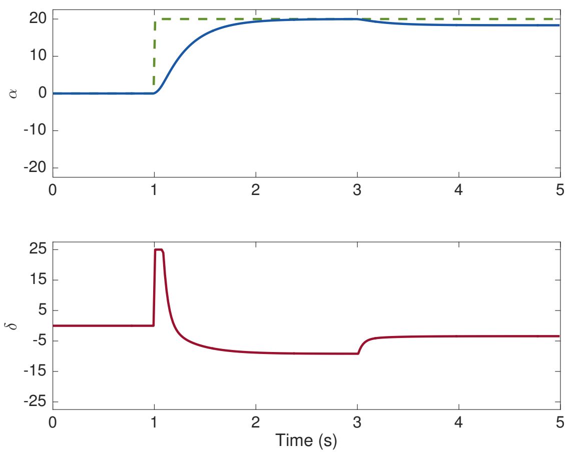
Trajectories in state-space ("phase plane").



Note treatment of infeasible reference request $\alpha = -20$.

Example cont'd

The effect of a constant disturbance that hits at $t = 3$:



Will need to add integral action (more on next lecture!)

Summary

- Softening of constraints
- Terminal sets via explicit constraint checking
- Reference-tracking via pseudo-reference

Reading instructions: Lecture notes §5.2.3-5.3.