ME 234(b): Model Predictive Control

Anushri Dixit

Spring 2022

Slides adapted from

Berkeley ME231 (by F. Borrelli, M. Morari, C. Jones)

Caltech

Recap: Constrained Optimal Control

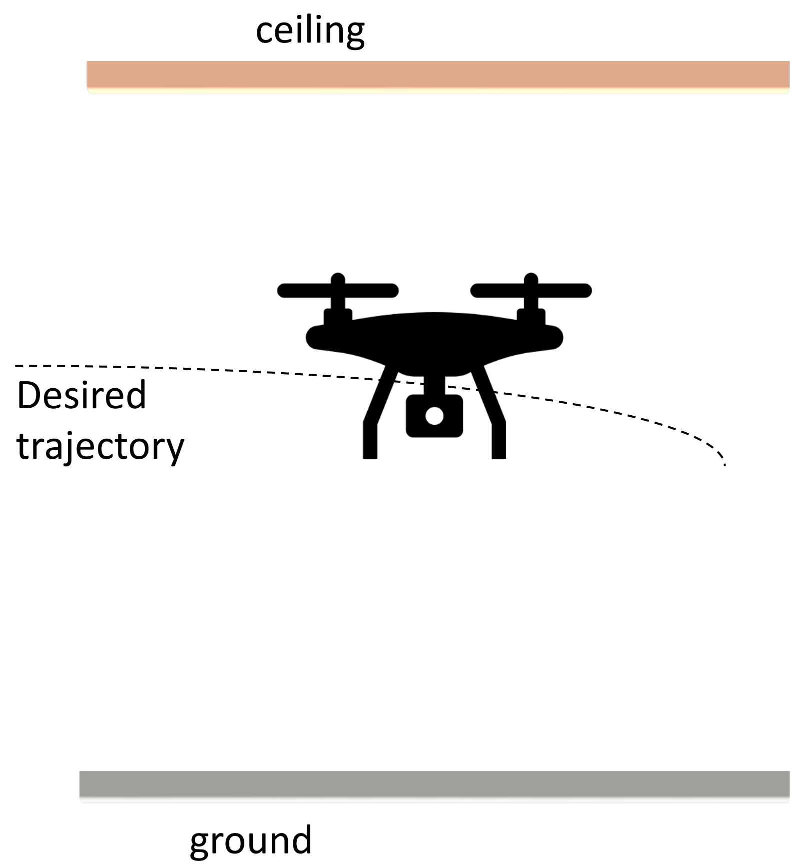
$$\min_{U_0} \quad x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right)$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U}$$

$$x_N \in \mathcal{X}_F$$

$$x_0 = x(0)$$



where, the state constraint set is

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} : F_x x \leq b_x\}$$

the control constraint set is

$$\mathcal{U} = \{u \in \mathbb{R}^{n_u} : F_u u \leq b_u\}$$

and the terminal set is given by

$$\mathcal{X}_F = \{x \in \mathbb{R}^{n_x} : F_f x \leq b_f\}$$

Caltech

Recap: Constrained Optimal Control

$$
\begin{aligned}
\min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U} \\
& x_N \in \mathcal{X}_F \\
& x_0 = x(0)
\end{aligned}
$$

We can solve the above problem using the Batch Approach

1. With substitution: $$\begin{aligned} J_0^*(x(0)) = \min_{U_0} \quad & (\mathcal{S}_x x(0) + \mathcal{S}_u U_0)^T \bar{Q} (\mathcal{S}_x x(0) + \mathcal{S}_u U_0) + U_0^T \bar{R} U_0 \\ \text{s.t.} \quad & G_0 U_0 \leq E_0 x(0) + w_0 \end{aligned}$$

2. Without substitution: $$\begin{aligned} J_0^*(x(0)) = \min_{X_0, U_0} \quad & \begin{bmatrix} X_0^T & U_0^T \end{bmatrix} \begin{bmatrix} \bar{Q} & 0 \\ 0 & \bar{R} \end{bmatrix} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \\ \text{s.t.} \quad & G_{0,\text{ineq}} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \leq E_{0,\text{ineq}} x(0) + w_{0,\text{ineq}} \\ & G_{0,\text{eq}} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} = E_{0,\text{eq}} x(0) \end{aligned}$$

Caltech

Receding Horizon Control

Ideally, we'd like to solve the constrained, infinite horizon optimal control problem,

$$
\begin{aligned}
J_0^*(x(0)) = \min_{u_0, u_1, \ldots} \quad & \sum_{k=0}^{\infty} \left(x_k^T Q x_k + u_k^T R u_k \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k, \quad \forall k \in \{0, 1, \ldots\} \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, \quad \forall k \in \{0, 1, \ldots\} \\
& x_0 = x(0)
\end{aligned}
$$

However, this problem has an infinite number of optimization variables, so we cannot compute it.

We, however, can solve the finite-horizon truncation of this problem.

Caltech

Receding Horizon Control

$$J_t^*(x(t)) = \min_{U_t} \; p(x_{t+N}) + \sum_{k=t}^{t+N-1} \left(x_k^T Q x_k + u_k^T R u_k \right)$$

$$\begin{aligned} \text{s.t.} \quad & x_{k+1} = A x_k + B u_k, && \forall k \in \{t, t+1, \ldots t+N-1\} \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U}, && \forall k \in \{t, t+1, \ldots t+N-1\} \\ & x_{t+N} \in \mathcal{X}_F \\ & x_t = x(t) \end{aligned}$$

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

The above optimization problem is a truncation of the infinite horizon problem wherein $p(x_{t+N}), \mathcal{X}_F$ approximate the remaining cost and the tail constraints, respectively.

Caltech

How to solve

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & p(x_{t+N}) + \sum_{k=t}^{t+N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{t+N} \in \mathcal{X}_F \\
& x_t = x(t)
\end{aligned}
\qquad \text{(OPT)}
$$

---

while $x_t \neq x_{\text{goal}}$ **do**
  Measure initial state at time t, $x_t = x(t)$
  Solve (OPT) to get the optimal control U_t
  if $U_t \neq \emptyset$ **then**
    Apply the first control input $U_t(1)$
  end if
  Wait for the new sampling time, $t = t + \Delta t$.
end while

Caltech

Notation

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & p(x_{t+N}) + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad && \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, && \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{t+N|t} \in \mathcal{X}_F \\
& x_{t|t} = x(t)
\end{aligned}
$$

We can now tell exactly what each state refers to:

- The input u at time k computed at time t: $u_{k|t}$
- The state x at time k computed at time t: $x_{k|t}$
- We apply the input $u_{t|t}$ at every time-step t.

Note that this can be extended to time-varying systems too!

Caltech

Implementation

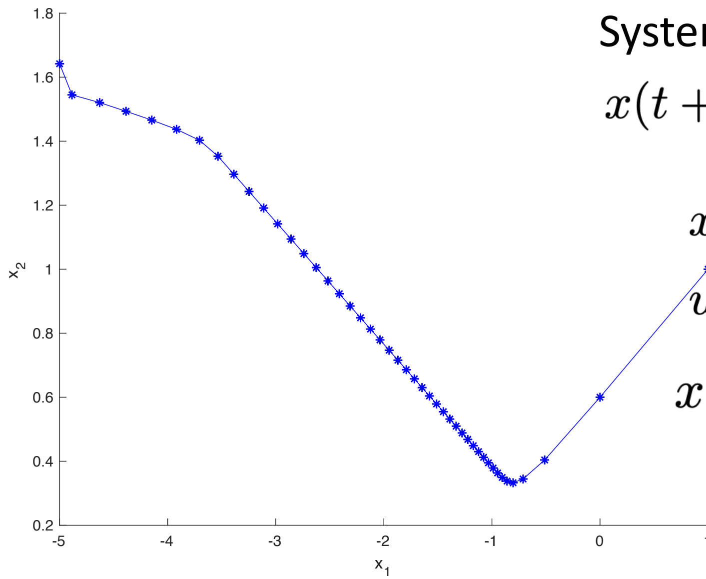
Let's look at how to implement this in MATLAB for a simple 2D system.

What we need:

- **YALMIP**: Makes it easy to set up optimization problems, well-documented, with a lot of tutorials
- **Solver**: YALMIP uses a variety of solvers. You need to have a solver that works best for your optimization problem.
- You can choose any other platform if you like!
  - Examples: CVX, MPT3, CasADi …

Caltech

Stability Issues: Horizon Length



N=12

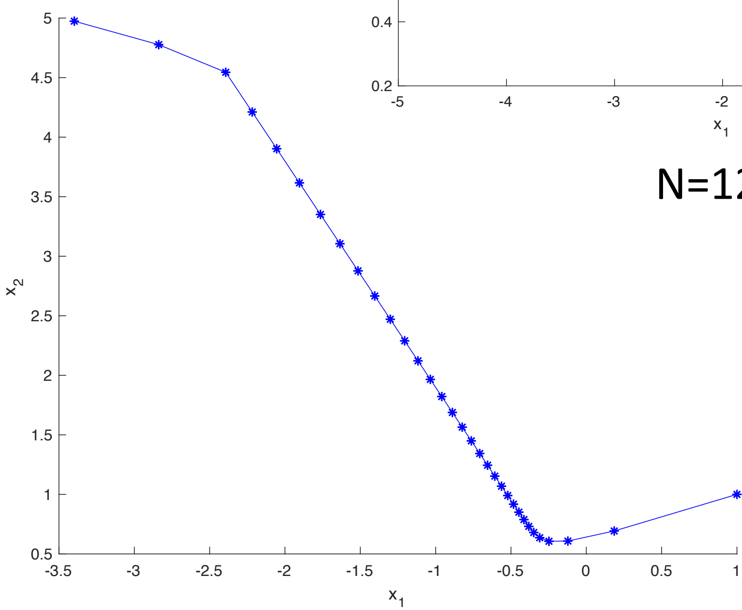
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

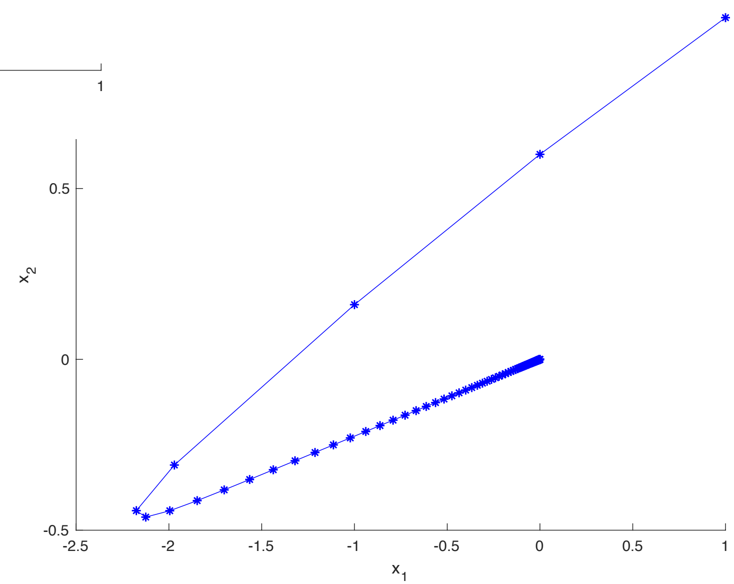
$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



N=2



N=50

Stability Issues: Cost weighting

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q_2 = 100 \begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}$$

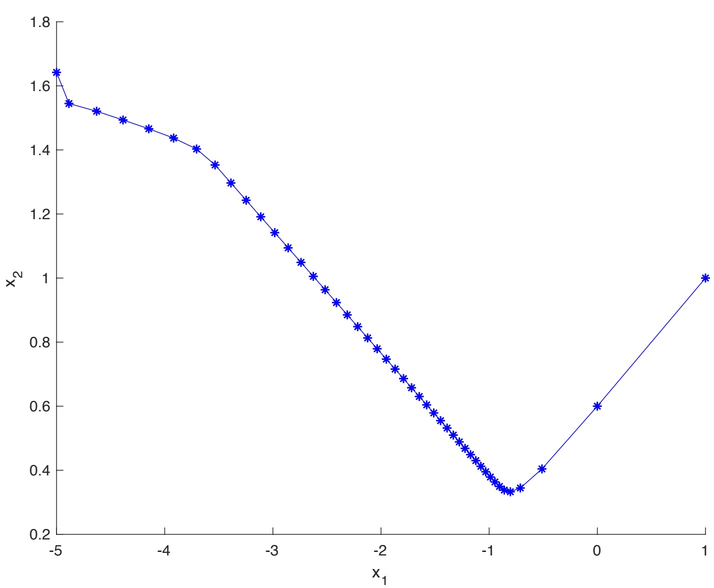
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

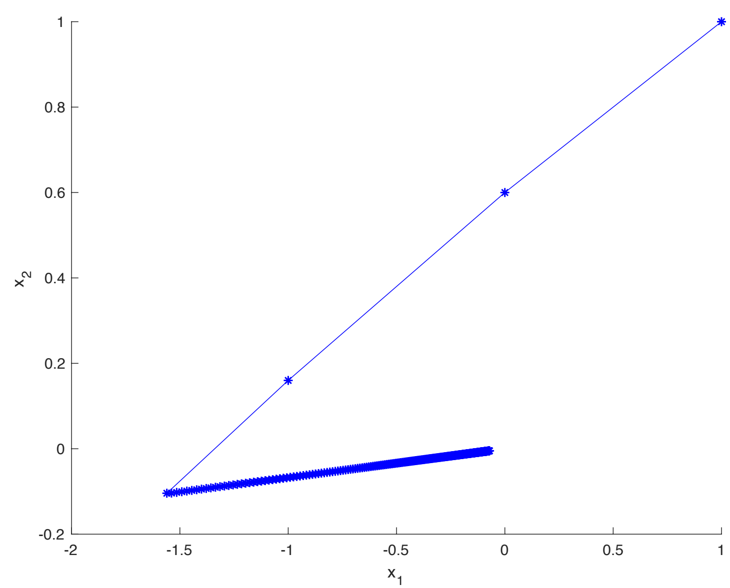
$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



Q=Q_1



Q = Q_2

Caltech

Feasibility Issues

Let's look at a specific case when the horizon N=12, $Q=Q_1$.

We see that the system does not converge to the origin (stability issues).

But perhaps if we allow for more iterations, the system will eventually converge to the origin.

Let's look at what happens after 200 time-steps instead of 100 steps.

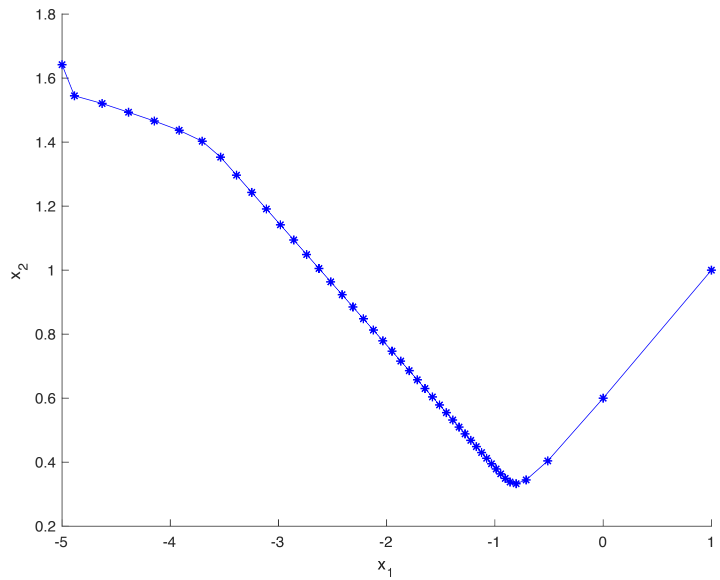
System:

$$x(t+1) = \begin{bmatrix}1 & 0\\0 & 1.1\end{bmatrix} x(t) + \begin{bmatrix}1\\0.5\end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix}-5 & 5\end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix}-1 & 1\end{bmatrix}$$

$$x(0) = \begin{bmatrix}1 & 1\end{bmatrix}^T$$



Feasibility Issues

Let's look at a specific case when N=12, Q=Q_1.

We see that the system does not converge to the origin.

But perhaps if we allow for more iterations, the system will eventually converge to the origin.

Let's look at what happens after 200 time-steps instead of 100 steps. Looks the same!

Unfortunately, the problem becomes infeasible early on, i.e., no solution can be obtained.

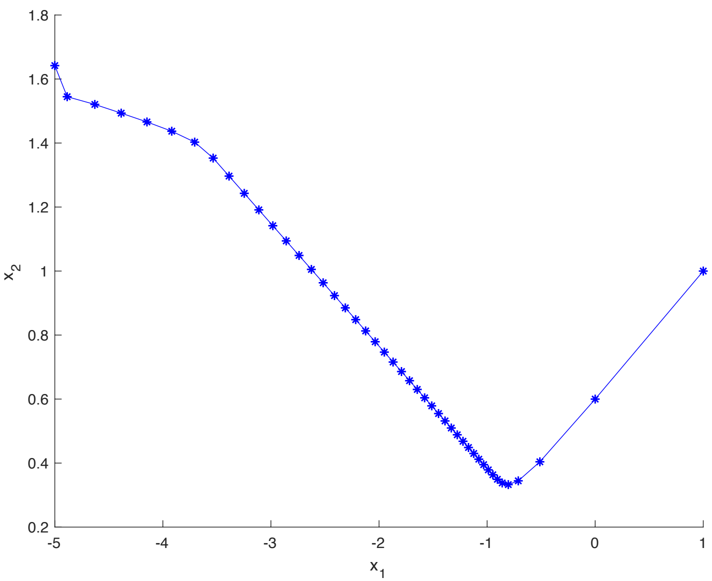
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



Caltech

Feasibility and Stability

More generally we have the following issues:

- Stability: System does not converge to the origin,
- Feasibility: MPC problem does not have a solution after some time-steps.

What can we do so that our MPC controller satisfies the above properties?

Feasibility and Stability

More generally we have the following issues:

- Stability: System does not converge to the origin,
- Feasibility: MPC problem does not have a solution after some time-steps.

What can we do so that our MPC controller satisfies the above properties?

Recall: MPC is a finite-horizon controller. These problems arise because of the “short-sightedness” of the controller.

We want to mimic an infinite-horizon controller. How?

Feasibility and Stability

$$J_t^*(x(t)) = \min_{U_t} \; p(x_{t+N}) + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t}\right)$$

$$\begin{aligned} \text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, && \forall k \in \{t, t+1, \ldots t+N-1\} \\ & x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, && \forall k \in \{t, t+1, \ldots t+N-1\} \\ & x_{t+N|t} \in \mathcal{X}_F \\ & x_{t|t} = x(t) \end{aligned}$$

We need a good terminal cost and terminal constraints.

Goal:

- **Stability**: Converge to the origin ⇒ cost is always decreasing across the iterations,
- **Recursive feasibility**: If the optimization is feasible at the first iteration, it is always feasible ⇒ existence of a feasible control input for all time-steps when starting at a feasible initial condition.

What's the simplest condition that will fix our MPC?

Caltech

Terminal set, $\mathcal{X}_F = \{0\}$

Let's look at the code again.

Earlier we saw that when N=12, $Q=Q_1$, the solution made the system unstable and not recursively feasible.

Let's add the terminal constraint,

$$x_{t+N|t} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

What happens?

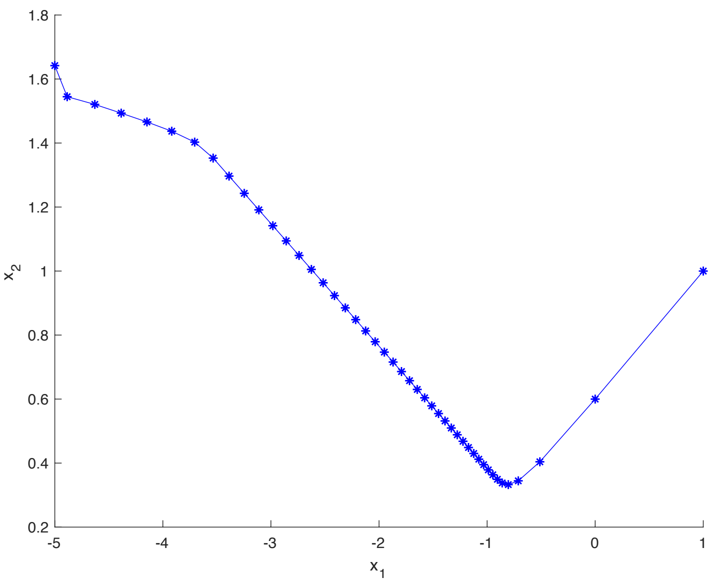
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



Caltech

Terminal set, $\mathcal{X}_F = \{0\}$

Let's look at the code again.

Earlier we saw that when N=12, Q=Q_1, the solution made the system unstable and not recursively feasible.

Let's add the terminal constraint,

$$x_{t+N|t} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

What happens?

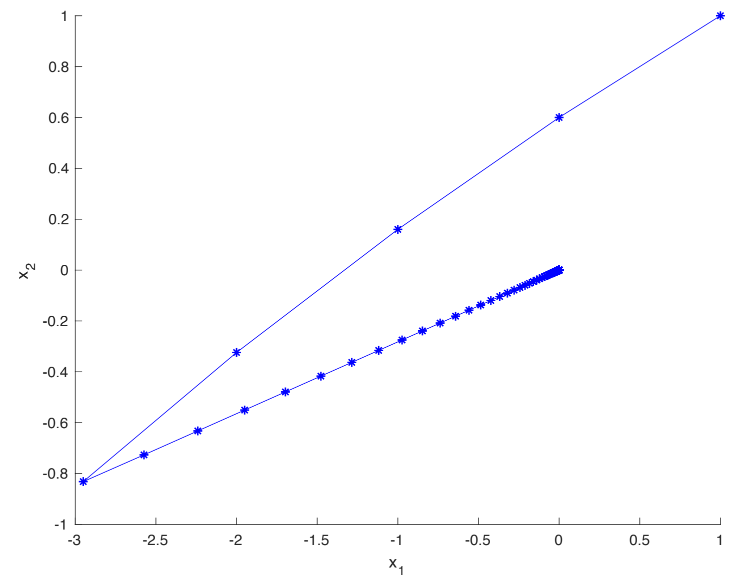
Stability and Recursive feasibility is attained!
How?

System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$



Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{t+N|t} = \mathbf{0}_{n_x}
\end{aligned}
$$

Recursive Feasibility: Let the solution of the optimization at time **t = 0**, be $U_0^* = \{u_{0|0}, u_{1|0}, \dots, u_{N-1|0}\}$. We apply the first control input.

Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{t+N|t} = \mathbf{0}_{n_x}
\end{aligned}
$$

Recursive Feasibility: Let the solution of the optimization at time t = 0, be $U_0^* = \{u_{0|0}, u_{1|0}, \dots, u_{N-1|0}\}$. We apply the first control input.

At the next iteration when, **t = 1** (or more generally, t = Δt), we know there exists a solution to the optimization problem. One such solution at time t=1, is $U_1 = \{u_{1|0}, u_{2|0}, \dots, u_{N-1|0}, 0\}$.

And so on…

Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

Recursive Feasibility

If the optimization is feasible at time $t = 0$, it is feasible for all future time steps.

Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{t+N|t} = \mathbf{0}_{n_x}
\end{aligned}
$$

Stability: Let the solution of the optimization at time t = 0, be $U_0^* = \{u_{0|0}, u_{1|0}, \ldots, u_{N-1|0}\}$. We apply the first control input.

At the next iteration when, t = 1 (or more generally, t = Δt), we know there exists a solution to the optimization problem. One such solution at time t=1, is $U_1 = \{u_{1|0}, u_{2|0}, \ldots, u_{N-1|0}, 0\}$.

Can we show the cost of the optimization is decreasing across iterations?

Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

$$J_t^*(x(t)) = \min_{U_t} \quad x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right)$$

$$\text{s.t.} \quad x_{k+1|t} = A x_{k|t} + B u_{k|t}, \qquad \forall k \in \{t, t+1, \dots t+N-1\}$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \qquad \forall k \in \{t, t+1, \dots t+N-1\}$$

$$x_{t+N|t} = \mathbf{0}_{n_x}$$

Stability:

$$J_0^*(x(0)) = \sum_{k=0}^{N-1} \left(x_{k|0}^{*T} Q x_{k|0}^* + u_{k|0}^{*T} R u_{k|0}^* \right)$$

$$= x_{0|0}^{*T} Q x_{0|0}^* + u_{0|0}^{*T} R u_{0|0}^* + \sum_{k=1}^{N-1} \left(x_{k|0}^{*T} Q x_{k|0}^* + u_{k|0}^{*T} R u_{k|0}^* \right)$$

$$= x_{0|0}^{*T} Q x_{0|0}^* + u_{0|0}^{*T} R u_{0|0}^* + J_1(x_{1|0})$$

$$> J_1(x_{1|0}) \qquad (\text{if } x_{0|0}^* \neq 0)$$

$$\geq J_1^*(x_{1|0})$$

Caltech

Proof: Stability & Recursive feasibility for $\mathcal{X}_F = \{0\}$

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{t+N|t} = \mathbf{0}_{n_x}
\end{aligned}
$$

Stability: Hence, we can show that the cost is decreasing across iterations,

$$J_0^*(x(0)) \geq J_1^*(x_{1|0}) \geq J_2^*(x_{2|1}) \geq \ldots \geq 0$$

We can show that the system (and the cost) must eventually converge to 0.

⇒ **Asymptotic Stability**

Caltech

Terminal set $\mathcal{X}_F = \{0\}$

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{t+N|t} = \mathbf{0}_{n_x}
\end{aligned}
$$

We showed that the above MPC is recursively feasible and stable!

Why is the terminal set {0} impractical?
The set of initial conditions from which the optimization is feasible at all is very small.

How do we improve this?

General form of terminal set and terminal cost

- We want to design a terminal set such that once we're inside this terminal set, there is always a way to get to the origin while satisfying constraints.
- Our terminal cost should be such that it gives us information on the true cost to reach the origin once we're in the terminal set.

To achieve this, we first need to understand set invariance.
Consider a control law, $u = Fx$.
Now the closed-loop system is expressed as:

$$x(t+1) = (A + BF)x(t)$$

Invariant set: Consider an autonomous system $x(t+1) = f(x(t))$.
The set $\mathcal{C}$ is invariant if $x(t) \in \mathcal{C} \implies f(x(t)) \in \mathcal{C}$.

Let's go back to our 2D example!

Caltech

Control invariant sets

Invariant set: Consider an autonomous system $x(t+1) = f(x(t))$. The set $\mathcal{C}$ is invariant if $x(t) \in \mathcal{C} \implies f(x(t)) \in \mathcal{C}$.

Let's go back to our 2D example!

System:

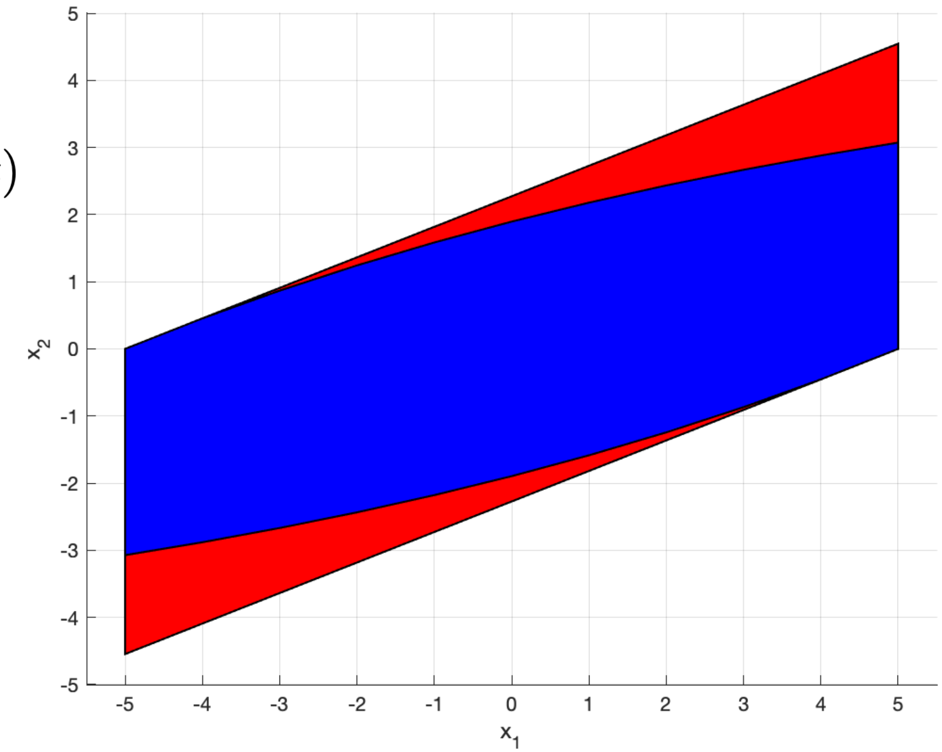
$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

Blue region: Invariant set with control constraints.

Red region: Invariant set in the absence of control constraints.



Caltech

General form of terminal set and terminal cost

For the 2D example, let's use the control invariant set as our **terminal set**.

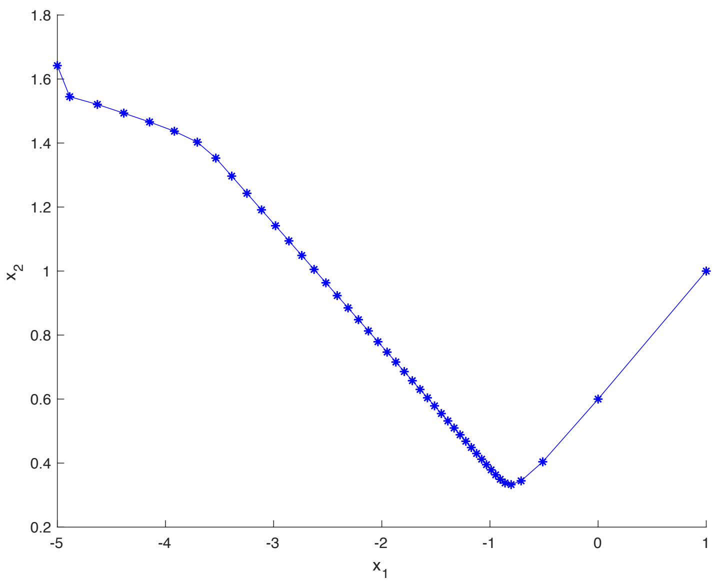
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



Without terminal set

N=12, Q=Q_1

Caltech

General form of terminal set and terminal cost

For the 2D example, let's use the control invariant set as our **terminal set**.

Remains in the control invariant set and is recursively feasible (# iterations = 1000).

Intuition: If the terminal set is control invariant, the MPC controller is recursively feasible. (we will prove this in the next couple of slides).

No convergence to the origin!

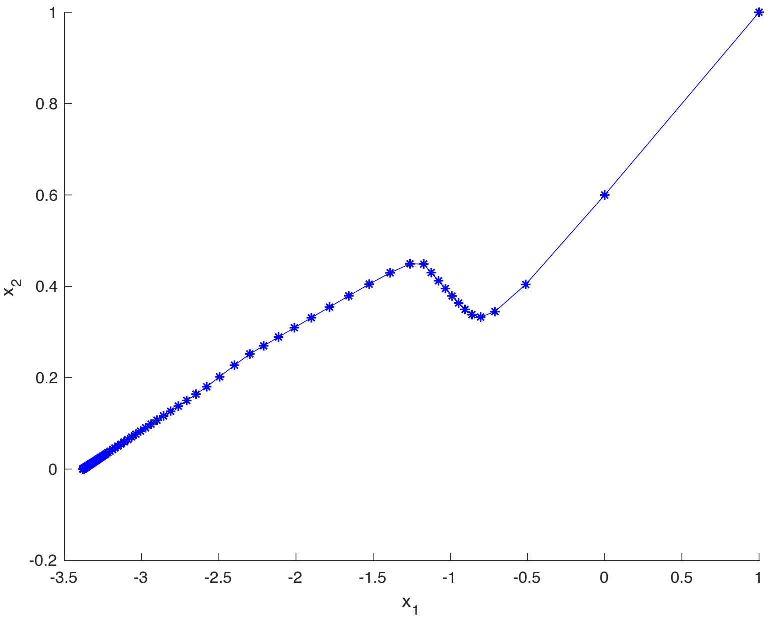
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



With invariant terminal set

N=12, $Q=Q_1$

Caltech

General form of terminal set and terminal cost

Theorem:

Consider the closed-loop system under the MPC control law $u^*_{t|t}$. The closed-loop system is asymptotically stable and the MPC optimization is recursively feasible if the following statements hold:

1. Stage cost is positive-definite (0 at the origin, strictly positive everywhere else),

Caltech

General form of terminal set and terminal cost

Theorem:

Consider the closed-loop system under the MPC control law $u^*_{t|t}$.
The closed-loop system is asymptotically stable and the MPC optimization is recursively feasible if the following statements hold:

1. Stage cost is positive-definite (0 at the origin, strictly positive everywhere else),
2. The sets $\mathcal{X}, \mathcal{X}_F, \mathcal{U}$ are closed and contain the origin,
3. The terminal set $\mathcal{X}_F \subseteq \mathcal{X}$ is control invariant.

Caltech

General form of terminal set and terminal cost

Theorem:

Consider the closed-loop system under the MPC control law $u^*_{t|t}$.
The closed-loop system is asymptotically stable and the MPC optimization is recursively feasible if the following statements hold:

1. Stage cost q is positive-definite (0 at the origin, strictly positive everywhere else),
2. The sets $\mathcal{X}, \mathcal{X}_F, \mathcal{U}$ are closed and contain the origin,
3. The terminal set $\mathcal{X}_F \subseteq \mathcal{X}$ is control invariant.
4. Terminal cost p is decreasing over time & satisfies the following condition wherein p, q are continuous and positive-definite functions (terminal cost is a continuous Lyapunov function)

$$\min_{v \in \mathcal{U}, Ax+Bu \in \mathcal{X}_F} (p(Ax+Bu) - p(x) + q(x,v)) \leq 0, \quad \forall x \in \mathcal{X}_F.$$

Terminal cost

Stage cost (we have assumed it to be quadratic $x^T Qx + u^T Ru$)

Caltech

Terminal set and cost for a LTI system with a quadratic cost

Consider the unconstrained linear OCP with a quadratic cost that we looked at in the previous lectures (also called the Linear-Quadratic regulator). We saw that the infinite-horizon control law is solved through the recursive approach

$$u^*(k) = -(B^T P_\infty B + R)^{-1} B^T P_\infty A x(k)$$

$$\triangleq F_\infty x(k).$$

where,

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

Caltech

Terminal set and cost for a LTI system with a quadratic cost

Consider the unconstrained linear OCP with a quadratic cost that we looked at in the previous lectures (also called the Linear-Quadratic regulator). We saw that the infinite-horizon control law is solved through the recursive approach

$$\begin{aligned} u^*(k) &= -(B^T P_\infty B + R)^{-1} B^T P_\infty A x(k) \\ &\triangleq F_\infty x(k). \end{aligned}$$

where,

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

We choose the **terminal weight**, $P = P_\infty$.

Caltech

Terminal set and cost for a LTI system with a quadratic cost

Consider the unconstrained linear OCP with a quadratic cost that we looked at in the previous lectures (also called the Linear-Quadratic regulator). We saw that the infinite-horizon control law is solved through the recursive approach

$$u^*(k) = -(B^T P_\infty B + R)^{-1} B^T P_\infty A x(k)$$

$$\triangleq F_\infty x(k).$$

where,

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

We choose the terminal weight, $P = P_\infty$.

We choose the **terminal set** to be the biggest invariant set (or the maximal invariant set) for the closed-loop system with the above control law designed for an infinite-horizon, unconstrained OCP.

$$(x_{k+1}, u_k) = \big(Ax_k + BF_\infty x_k, F_\infty x_k\big) \in \mathcal{X}_F \times \mathcal{U}, \quad \forall x_k \in \mathcal{X}_F \subseteq \mathcal{X}$$

Caltech

Proof Sketch: MPC Stability & Recursive feasibility

$$J_t^*(x(t)) = \min_{U_t} \quad x_{t+N|t}^T P_\infty x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right)$$

$$\text{s.t.} \quad x_{k+1|t} = A x_{k|t} + B u_{k|t}, \qquad \forall k \in \{t, t+1, \ldots t+N-1\}$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \qquad \forall k \in \{t, t+1, \ldots t+N-1\}$$

$$x_{t+N|t} \in \mathcal{X}_F$$

Recursive Feasibility: Like we did in the $\mathcal{X}_F = \{0\}$ case, we start with the solution of the above optimization problem at time **t=0** given by $U_0^* = \{u_{0|0}^*, u_{1|0}^*, \ldots, u_{N-1|0}^*\}$. We apply the first control input.

Caltech

Proof Sketch: MPC Stability & Recursive feasibility

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P_\infty x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \qquad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \qquad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{t+N|t} \in \mathcal{X}_F
\end{aligned}
$$

Recursive Feasibility: Like we did in the $\mathcal{X}_F = \{0\}$ case, we start with the solution of the above optimization problem at time t=0 given by $U_0^* = \{u_{0|0}^*, u_{1|0}^*, \dots, u_{N-1|0}^*\}$. We apply the first control input.

At the next time-step, **t=1**, a candidate solution for the MPC optimization is $U_1 = \{u_{1|0}^*, u_{2|0}^*, \dots, u_{N-1|0}^*, F_\infty x_{N|0}^*\}$. The MPC is feasible with the solution $U_1^* = \{u_{1|1}^*, u_{2|1}^*, \dots, u_{N-1|1}^*, u_{N|1}^*\}$.

At the next time-step, we can similarly find a candidate solution based on the solution at the previous time-step and so on…

Caltech

Proof Sketch: MPC Stability & Recursive feasibility

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P_\infty x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t} \right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \quad \forall k \in \{t, t+1, \ldots t+N-1\} \\
& x_{t+N|t} \in \mathcal{X}_F
\end{aligned}
$$

Stability: We have established that a solution will always exist for the above MPC optimization. Can we show that the cost is decreasing?

Caltech

Proof Sketch: MPC Stability & Recursive feasibility

$$
\begin{aligned}
J_t^*(x(t)) = \min_{U_t} \quad & x_{t+N|t}^T P_\infty x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t}\right) \\
\text{s.t.} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}, \qquad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \qquad \forall k \in \{t, t+1, \dots t+N-1\} \\
& x_{t+N|t} \in \mathcal{X}_F
\end{aligned}
$$

Stability: Consider the optimal cost at time t

$$J_t^*(x(t)) = x_{t+N|t}^{*T} P_\infty x_{t+N|t}^* + \sum_{k=t}^{t+N-1} \left(x_{k|t}^{*T} Q x_{k|t}^* + u_{k|t}^{*T} R u_{k|t}^*\right)$$

We know that at time t+1, a candidate solution is given by $U_{t+1} = \{u_{t+1|t}^*, u_{t+2|t}^*, \dots, u_{t+N-1|t}^*, F_\infty x_{t+N|t}^*\}$. So we have the suboptimal cost,

$$
\begin{aligned}
J_{t+1}(x(t+1)) = (A x_{t+N|t}^* + B u_{t|t+1})^T P_\infty (A x_{t+N|t}^* + B u_{t|t+1}) + \\
x_{t+N|t}^{*T} Q x_{t+N|t}^* + u_{t|t+1}^T R u_{t|t+1} + \sum_{k=t+1}^{t+N-1} \left(x_{k|t}^{*T} Q x_{k|t}^* + u_{k|t}^{*T} R u_{k|t}^*\right)
\end{aligned}
$$

Caltech

Proof Sketch: MPC Stability & Recursive feasibility

$$J_t^*(x(t)) = \min_{U_t} \quad x_{t+N|t}^T P_\infty x_{t+N|t} + \sum_{k=t}^{t+N-1} \left(x_{k|t}^T Q x_{k|t} + u_{k|t}^T R u_{k|t}\right)$$

$$\text{s.t.} \quad x_{k+1|t} = A x_{k|t} + B u_{k|t}, \qquad \forall k \in \{t, t+1, \dots t+N-1\}$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \qquad \forall k \in \{t, t+1, \dots t+N-1\}$$

$$x_{t+N|t} \in \mathcal{X}_F$$

Stability: We can show that the cost is decreasing across each iteration. Once we show decreasing cost, we can also show stability.

$$J_t^*(x_t) - J_{t+1}^*(x_{t+1}) \geq J_t^*(x_t) - J_{t+1}(x_{t+1})$$

$$= x_{t+N|t}^{*T} P_\infty x_{t+N|t}^* + x_{t|t}^{*T} Q x_{t|t}^* + u_{t|t}^{*T} R u_{t|t}^* -$$

$$(A x_{t+N|t}^* + B u_{t|t+1})^T P_\infty (A x_{t+N|t}^* + B u_{t|t+1}) -$$

$$x_{t+N|t}^{*T} Q x_{t+N|t}^* - u_{t|t+1}^T R u_{t|t+1}$$

$$= x_{t|t}^{*T} Q x_{t|t}^* + u_{t|t}^{*T} R u_{t|t}^*$$

$$> 0 \qquad (= 0 \text{ when } x_{t|t} = 0, u_{t|t} = 0)$$

Summary: Example

For the 2D example, let's use the control invariant set as our **terminal set** and the **terminal cost** given by using the Algebraic Riccati equation.

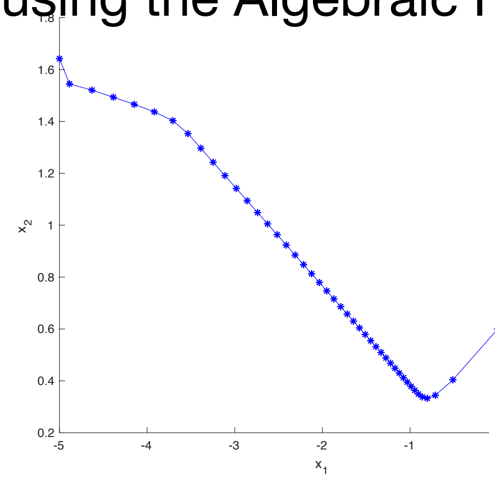
System:

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t)$$

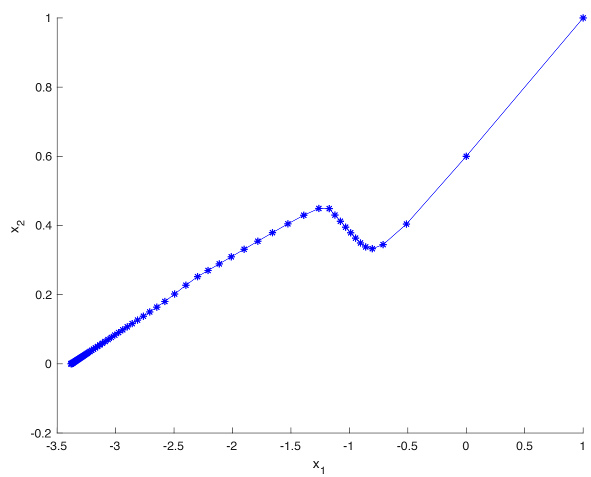
$$x(t) \in \begin{bmatrix} -5 & 5 \end{bmatrix}^2$$

$$u(t) \in \begin{bmatrix} -1 & 1 \end{bmatrix}$$

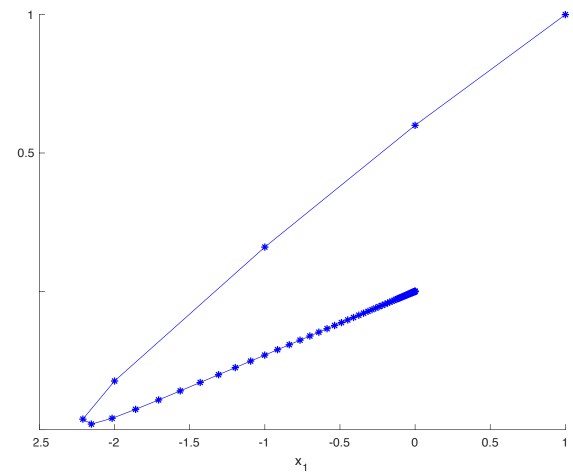
$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$$



Without terminal set, terminal constraint N=12, $Q=Q_1$



With invariant terminal set only N=12, $Q=Q_1$



With invariant terminal set and terminal constraint N=12, $Q=Q_1$

Caltech

Summary

- We looked at the MPC optimization and analyzed its closed-loop properties: stability and recursive feasibility. We tried to mimic an infinite-horizon controller to get these properties.
- Recursive feasibility is obtained if we have a terminal set that is invariant. However, terminal sets are not often used in practice,
  - These sets are hard to compute,
  - They reduce the size of the set of initial conditions from which the MPC provides a solution, also called the region of attraction.
  - The {0} terminal set makes the region of attraction even smaller.
- Stability is obtained with terminal costs that are Lyapunov functions.
- Often, we can get away with just making the length of the horizon longer.

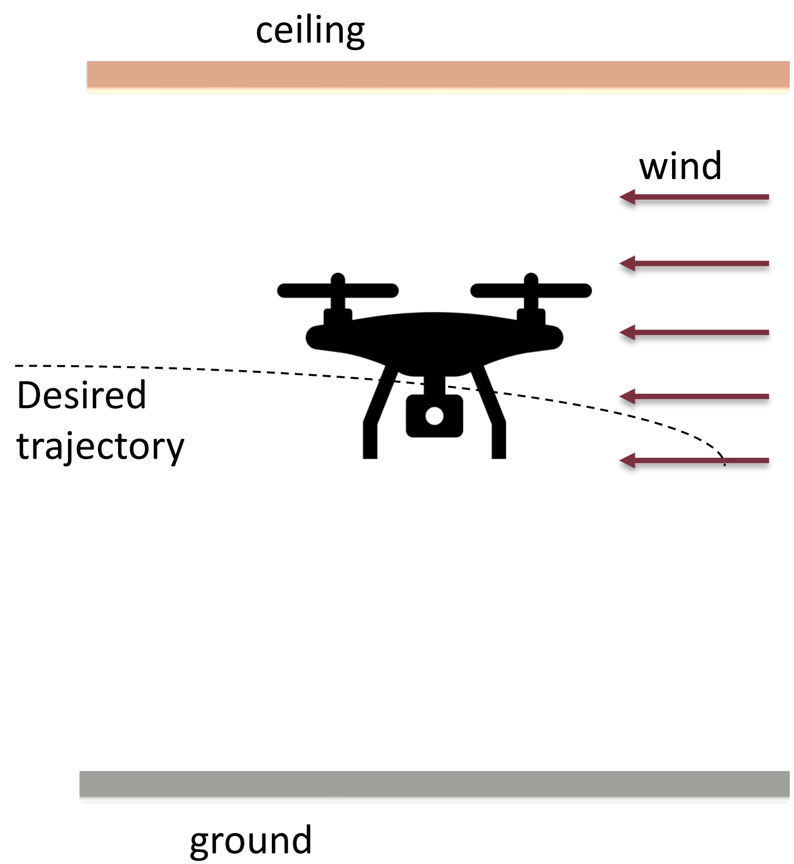
Caltech

What's next?

Consider a drone in hover.

We want it to:

1. Track a desired trajectory, ☑
2. While not crashing into the ceiling or the ground, ☑
3. And account for disturbances.



Caltech