Lecture 13: Robust MPC

I. Uncertainty Models and Robust MPC Formulation

- **Initial Remarks: Why Robust MPC**
  1. MPC relies on a model, but models are far from perfect
  2. Noise and model inaccuracies can cause: Constraint Violation or Sub-optimal Behavior
  3. Persistent noise prevents the system from converging to a single point

Therefore, we need to incorporate some noise models into the MPC formulation, but note that there is no such thing as the best way to deal with it. Solving the resulting optimal control problem is extremely difficult. Many approximations would be made, but most are very conservative

Predictive Control Assumption	The Real World
$x^+ = f(x,u)$ System evolves in a predictable fashion	$x^+ = g(x,u,w,\theta)$ Random noise w changes the evolution of the system The model structure is unknown Unknown parameters θ impact the dynamics

Note: Here we have two different kinds of uncertainty. The unknown parameter θ is constant, meaning it can be learned/identified, while the random noise w is stochastic, meaning it changes over time and can not be learned.

This lecture: What can we hope to do in this (real) situation?

- **Robust Constrained Control Formulation**

Goals of Robust Constrained Control

Consider the uncertain constrained system:

$$x^+ = f(x,u,w,\theta) \quad (x,u) \in \mathcal{X},\ \mathcal{U},\ w \in \mathbb{W},\ \theta \in \Theta$$

We want to design a control law $u = \kappa(x)$ such that the system:

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

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

Idea: we design a control law that will satisfy constraints and stabilize the system for all possible disturbances! That's why we choose to optimize (expected/worst-case) "performance"

Meeting these goals requires some knowledge/assumptions about the random values w and θ. Therefore, we now introduce some typical uncertainty models.

- **Common Uncertainty Models**

1. Measurement / Input Bias

$$g(x,u,w,\theta)=f(x,u)+\theta$$

where θ is unknown but a constant

The unexpected offset can cause constraint violation

Offset doesn't change, or changes slowly with time: we can generally handled by estimating offset and compensating (lecture 10 tracking without offset)

Constraint violation still possible before the offset is estimated.

2. Linear Parameter Varying System

$$g(x,u,w,\theta)=\sum_{k=0}^{t}\theta_k A_k x+\sum_{k=0}^{t}\theta_k B_k u \qquad \mathbf{1}^{\mathrm{T}}\theta=1,\ \theta\geq 0$$

A_k, B_k known, θ_k unknown, but constant, here t means the number of possible cases, not time.

Actual system is linear, but exact dynamics are unknown

Preventing constraint violation requires considering all possible trajectories (very conservative), since only one of the cases is happening but we consider all.

Often handled by estimating θ: **adaptive control**, since it is constant, or changes slowly. It is very difficult to handle if the system is unstable

3. Polytopic Uncertainty

$$g(x,u,w,\theta)=\sum_{k=0}^{t}w_k Ax+\sum_{k=0}^{t}w_k B_k u \qquad \mathbf{1}^{\mathrm{T}}w=1,\ w\geq 0$$

A_k, B_k known, w_k unknown and is changing at each sample time, here t means the number of possible cases, not time.

Dynamics change randomly at each point in time $\Rightarrow$ essentially nonlinear system

Preventing constraint violation requires considering all possible trajectories (not conservative, since they all can happen)

Commonly dealt with **via robust MPC**

We will not cover this case in this course, but the analysis is similar to additive noise.

4. Additive Stochastic Noise

$$g(x,u,\theta,w)=Ax+Bu+w$$

where the distribution of w is known

Problem is significantly more challenging (even to formulate the goals), still a topic of active research

5. Additive Bounded Noise

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

where A, B are known, w is unknown and challenging with each sample time

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

We will mainly focus on this type of system in the robust MPC lecture

II. Impact of Bounded Additive Noise

Now we consider the characteristic for our additive bounded noise model.

- **Uncertain State Evolution**

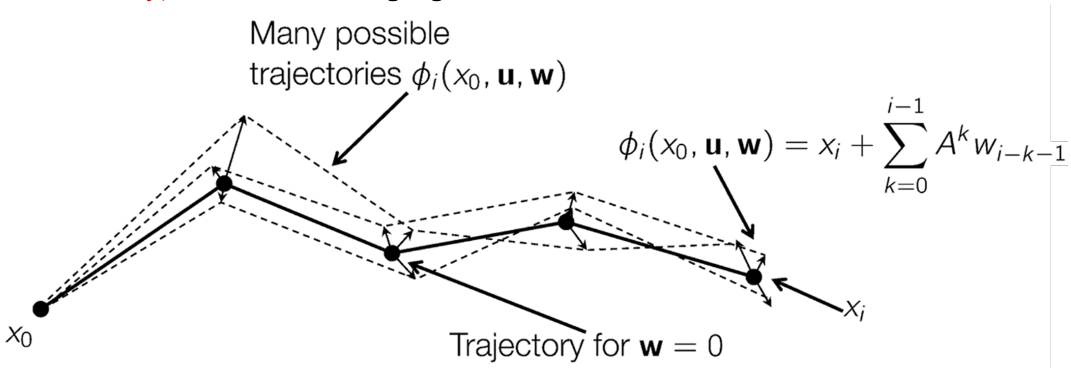
Given the current state x_0, the model $x^+ = Ax + Bu + w$ and the set $\mathbb{W}$, where can the state be i steps in the future?

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

Rolling out the dynamics, we have:

Nominal System	Uncertain System
$x^+ = Ax + Bu$	$x^+ = Ax + Bu + w,\ w \in \mathbb{W}$
$x_1 = Ax_0 + Bu_0$ $x_2 = A^2 x_0 + ABu_0 + Bu_1$ $\vdots$ $x_i = A^i x_0 + \sum_{k=0}^{i-1} A^k B u_{i-k-1}$	$\phi_1 = Ax_0 + Bu_0 + w_0$ $\phi_2 = A^2 x_0 + ABu_0 + Bu_1 + Aw_0 + w_1$ $\vdots$ $\phi_i = A^i x_0 + \sum_{k=0}^{i-1} A^k B u_{i-k-1} + \sum_{k=0}^{i-1} A^k w_{i-k-1}$ $\Rightarrow \phi_i = x_i + \sum_{k=0}^{i-1} A^k w_{i-k-1}$

Conclusion: uncertain evolution is the nominal system + offset caused by the disturbance (this result follows from linearity). As the following figure shows:



- **Defining a Cost to Minimize**

Previously, we defined some function that describes a 'good' trajectory:

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

The cost definition is for a single trajectory. However, there are now many trajectories that may occur, depending on the disturbance $\mathbf{w}$, so we need to come up with a way to define the cost.

Firstly, we can write out the cost function with disturbance, note that it is now a function of the disturbance seen, and therefore each possible trajectory has a different cost:

$$J(x_0,\mathbf{u},\mathbf{w})=\sum_{i=0}^{N-1} l(\phi_i(x_0,\mathbf{u},\mathbf{w}),u_i)+V_f(\phi_N(x_0,\mathbf{u},\mathbf{w}))$$

We need to 'eliminate' the dependence on $\mathbf{w}$. Therefore several ways to do so:

1. Minimize the expected value (requires some assumption on the distribution)
$$V_N(x_0,\mathbf{u})=\mathbb{E}[J(x_0,\mathbf{u},\mathbf{w})]$$
2. Minimize the variance (requires some assumption on the distribution)
$$V_N(x_0,\mathbf{u})=\mathrm{Var}[J(x_0,\mathbf{u},\mathbf{w})]$$
3. Take the worst case
$$V_N(x_0,\mathbf{u})=\max_{\mathbf{w}\in\mathbb{W}^{N-1}}[J(x_0,\mathbf{u},\mathbf{w})]$$
4. Take the nominal case
$$V_N(x_0,\mathbf{u})=J(x_0,\mathbf{u},0)$$

In this lecture we will **assume the nominal case** for simplicity.

We will 'fluff' over the stability proof, because we cannot demonstrate robust stability in this case (i.e., asymptotic convergence for all possible disturbances).

We will introduce a new notion of stability later that will allow us to analyze this case.

Note: recall the goals of robust constraint control proposed at the beginning of the lecture. In this part, we are proposing methods to deal with the third goal- Optimizes (expected/worst-case) "performance".

- **Robust Constraint Satisfaction: Overview**

Recall: we break the MPC prediction into two parts:

Part A: During the prediction horizon:

$$\begin{cases}\phi_{i+1}=A\phi_i+Bu_i+w_i\\ u_i\in\mathcal{U}\\ \phi_i\in\mathcal{X},\ \forall\mathbf{w}\in\mathbb{W}^N\end{cases}$$

$i=0,\ldots,N-1$

Optimize over control actions $\{u_0,\ldots,u_{N-1}\}$

Enforce constraints explicitly by imposing $\phi_i\in\mathcal{X}$ and $u_i\in\mathcal{U}$ for all disturbance sequences $\mathbf{w}$

Part B: After the prediction horizon:

$$\begin{cases}\phi_{i+1}=(A+BK)\phi_i+w_i\\ \phi_N\in\mathcal{X},\ \forall w_i\in\mathbb{W}\end{cases}$$

$i=N,\ldots$

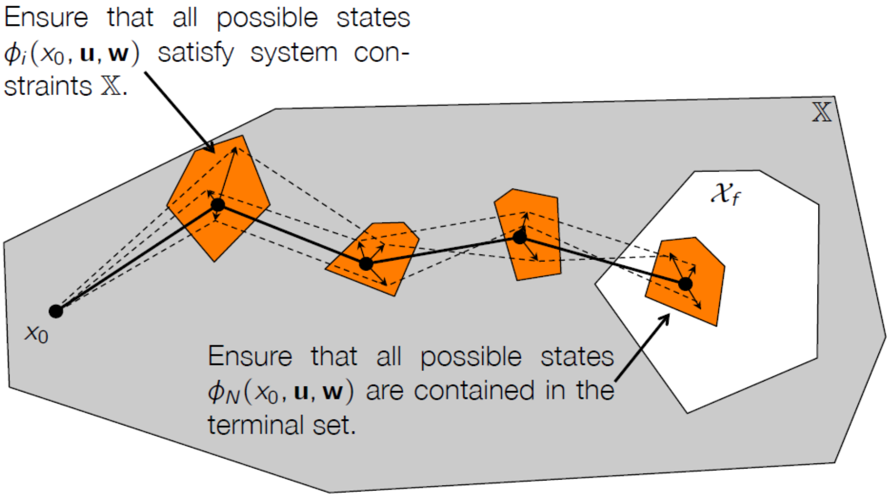
Assume control law to be linear $u_i=K\phi_i$

Enforce constraints explicitly by imposing ϕ_N to be in a robust invariant set $\mathcal{X}_f\subseteq\mathcal{X}$ and $K\mathcal{X}_f\in\mathcal{U}$ for the system $\phi_{i+1}=(A+BK)\phi_i+w_i$, for any noise w_i

Therefore, in the following sections, we will discuss:

1. Robustly ensuring constraints of the sequence $\phi_1, \ldots, \phi_{N-1}$ (for Part A, consider a sequence of a bundle of the state evolution sequences with all possible disturbances over the prediction horizon)
2. Robustly enforcing constraints of a linear system (for Part B, consider the linear system state evolution with disturbance, turning into the single-step robust forward invariance analysis)

The basic idea is: 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.



Note:

1. recall the goals of robust constraint control proposed at the beginning of the lecture, in this part (and the next two sections), we are proposing methods to deal with the first goal- Satisfies constraints: $\{x_i\} \subset \mathcal{X}_i$, $\{u_i\} \subset \mathcal{U}$ for all disturbance realizations
2. In the notation above, $\mathbb{W}^N = \underbrace{\mathbb{W} \times \mathbb{W} \ldots \times \mathbb{W}}_{N \text{ times}}$

   it is a simplified verision that assumes all the disturbances are from the same bounded set and we will use this setting in this lecture. In fact, disturbances at different timestep i can come from different set $\mathbb{W}_i$. Our methodology discussed below still holds for those more generalized cases.

- **Robust Constraint Satisfaction for Linear System: Robust Invariant Set**

Here we discuss how to deal with the second part (part B) first because it is easier and would lay the foundations for the discussions for part A. As mentioned before, invariance is the key for constraints satisfaction. Therefore, we analyze the robust constraint satisfaction, either for an autonomous system $x^+ = f(x, w)$, or for a closed-loop system $x = f(x, \kappa(x), w)$ with a given controller $\kappa(x)$, through the robust positive invariance.

Recall in Lecture 9, we introduced the concept of pre set for the robust case, as the following shows:

Definition (Robust Pre Set):

Given a set Ω and the dynamic system $x^+ = f(x, w)$, the pre set of Ω is is the set of states that evolve into the target set Ω in one time step **for all values of the disturbance** $w \in \mathbb{W}$:

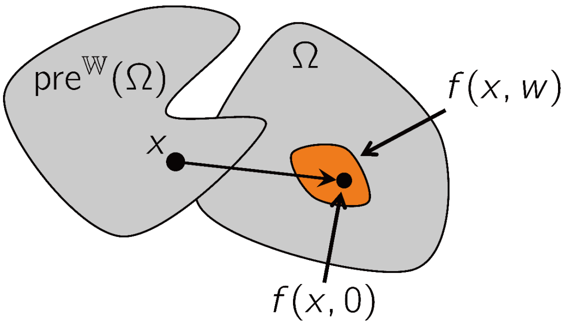
$$\operatorname{pre}^{\mathbb{W}}(\Omega)=\{x \mid f(x,w)\in\Omega,\ \forall w\in\mathbb{W}\}$$

And we also mentioned how to calculate the robust pre set for additive disturbance before. Given the system $f(x,w)=Ax+w$ and the set $\Omega=\{x \mid Fx\le f\}$, the robust pre set is:

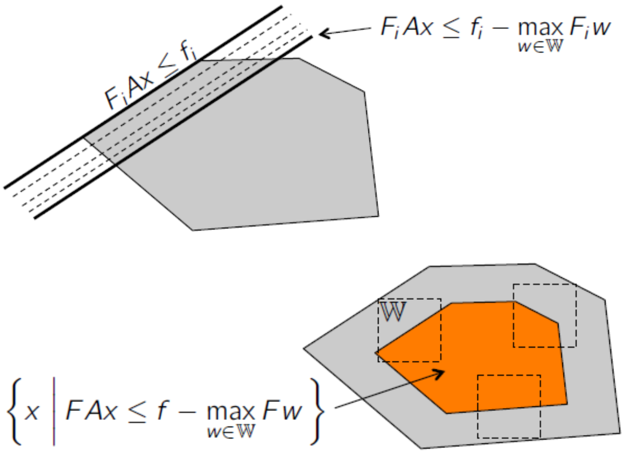
$$\operatorname{pre}^{\mathbb{W}}(\Omega)=\{x \mid Ax+w\in\Omega,\ \forall w\in\mathbb{W}\}=\{x \mid FAx+Fw\le f,\ \forall w\in\mathbb{W}\}$$

$$=\left\{x \mid FAx\le f-\max_{w\in\mathbb{W}} Fw\right\}=\{x \mid FAx\le f-h_{\mathbb{W}}(F)\}=A(\Omega\ominus\mathbb{W})$$

where $h_{\mathbb{W}}$ is the support function of F and $\ominus$ is the Pontryagin difference. The robust preset and its calculation can be illustrated by the following figures:



Robust Pre Set



Robust Pre Set Calculation

Now, we give the definition of the robust invariant set and conditions for robust invariance..

Definition (Robust Positive Invariant Set):

A set $\mathcal{O}^{\mathbb{W}}$ is said to be a robust positive invariant set for the autonomous system $x^{+}=f(x,w)$ if:

$$x\in\mathcal{O}^{\mathbb{W}} \Rightarrow f(x,w)\in\mathcal{O}^{\mathbb{W}},\ \forall w\in\mathbb{W}$$

Theorem (Geometric Condition for Robust Invariance):

A set Ω is a robust positive invariant set if and only if:

$$\mathcal{O}\subseteq\operatorname{pre}^{\mathbb{W}}(\mathcal{O})$$

And according to this condition we have the following algorithm

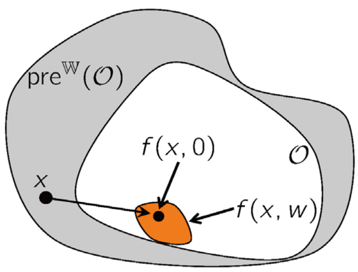
Algorithm (Algorithm to Compute Robust Invariant Set):

Input: f, $\mathcal{X}$, $\mathbb{W}$

(system dynamics, feasible set, and disturbance set)

Output: $\mathcal{O}_{\infty}^{\mathbb{W}}$

(maximum robust invariant set)



$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \operatorname{pre}^{\mathbb{W}}(\Omega_i)\cap\Omega_i$
  if $\Omega_{i+1}=\Omega_i$ **then**
    return $\mathcal{O}_{\infty}^{\mathbb{W}}=\Omega_i$
    end if
  end loop

Note: It can be observed that the robust invariant set calculation is the same as the nominal case, with $\operatorname{pre}(\Omega)$ replaced by $\operatorname{pre}^{\mathbb{W}}(\Omega)$

Example: Computing Robust Invariant Set

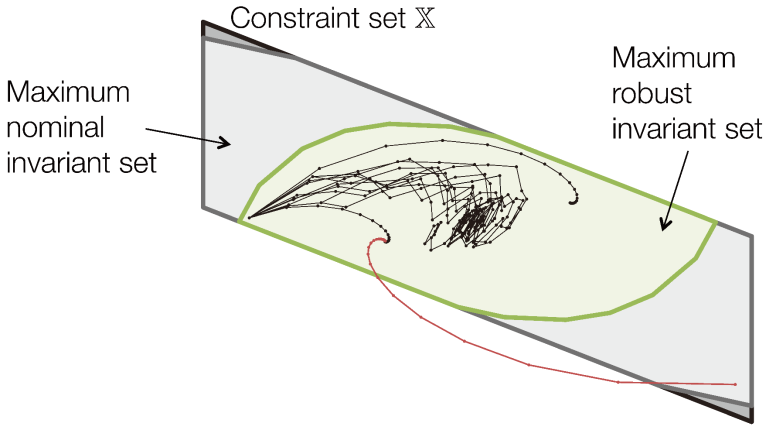
Consider the linear system with additive bounded disturbance:

$$x(A+BK)x+w,\ A=\begin{bmatrix}1&1\\0&1\end{bmatrix},\ B=\begin{bmatrix}1\\0.5\end{bmatrix}$$

where system constraints and disturbances sets are:

$$\mathcal{X}=\{x|\|x\|_\infty\leq 5, \|Kx\|_\infty\leq 1\},\ \mathbb{W}=\{w|\|w\|_\infty\leq 0.3\}$$

and K is the LQR gain for $Q=0.1I,\ R=1$, the calculated robust invariant set is shown below:



Note: In the plot above, we can see that an initial state that is in the nominal invariant set (red) might be pushed out of the constraint set due to the existence of disturbances. Only the state starts from the robust invariants set would not violate the constraints under disturbances.

- **Robust Constraint Satisfaction for MPC Sequences** $\phi_1,\ldots,\phi_{N-1}$

As mentioned above, the uncertain system evolution is:

$$\phi_i(x_0,\mathbf{u},\mathbf{w})=\left\{x_i+\sum_{k=0}^{i-1}A^k w_{i-k-1}\,\middle|\,\mathbf{w}\in\mathbb{W}^i\right\}$$

Our goal is: Ensure that constraints are satisfied for the MPC sequence, and we hope to express things in the nominal system x_i because the core idea is as long as the nominal system satisfies the constraints, then the uncertain system will too.

Therefore, assume that $\mathcal{X}=\{x|\ Fx\leq f\}$, then our goal is equivalent to:

$$Fx_i+F\sum_{k=0}^{i-1}A^k w_{i-k-1}\leq f,\ \forall\mathbf{w}\in\mathbb{W}^i$$

We've seen this before while computing the robust pre-set, it turns into calculating the support function of F and the support function can be pre-computed offline:

$$Fx_i\leq f-\max_{\mathbf{w}\in\mathbb{W}^i}F\sum_{k=0}^{i-1}A^k w_{i-k-1}=f-h_{\mathbb{W}^i}\left(F\sum_{k=0}^{i-1}A^k w_{i-k-1}\right)$$

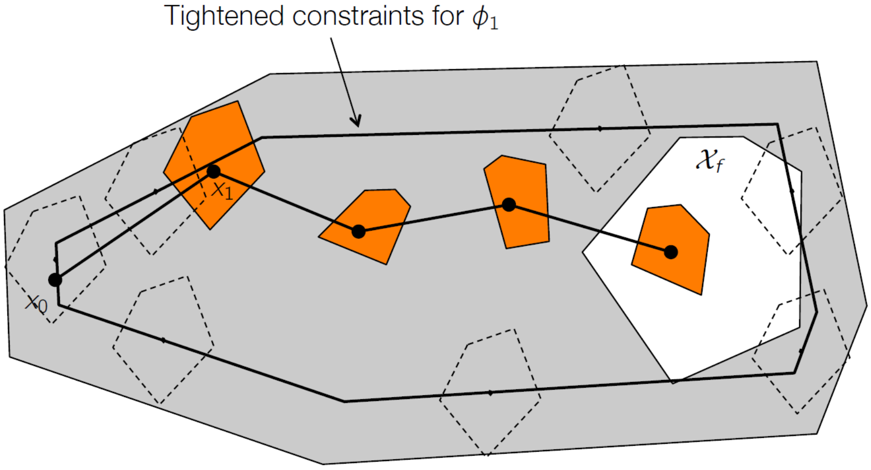
Therefore, we have:

Ensure $\phi_i\in\mathcal{X},\ \forall\mathbf{w}\in\mathbb{W}^N\Rightarrow$ Require $x_i\in\mathcal{X}\ominus[I\quad A\quad\ldots\quad A^{i-1}]\mathbb{W}^i$ (recall the support function and its relationship with Pontryagin Difference)

In addition, we also need to ensure that the N -th state $\phi_N(x_0,\mathbf{u},\mathbf{w})$ is contained in the robust control invariant set $\mathcal{O}_\infty^{\mathbb{W}}$:

$$\phi_N(x_0,\mathbf{u},\mathbf{w}) \subseteq \mathcal{X}_f = \mathcal{O}_\infty^{\mathbb{W}}$$

A demonstration of constraint tightening for ϕ_i is shown below:



Note:

1. In the above formulation, we can see that for different timestep i, the constraint tightening is different, and under our setting (all the disturbances sets at each timestep is the same), it is clear that the constraints satisfaction would be tighter and tighter as time goes b. This means if our horizon is long enough, in the end the feasible set would be empty and we can do nothing. Therefore, this is a very naïve version of robust constraint satisfaction and it needs to be improved.
2. As mentioned before, we can also assume that the disturbances from different step timestep i are from different constraints set $\mathbb{W}_i$, but we need to be careful about the index correspondence. At time i, w_0 has already been propagated by the dynamics $i-1$ times, so the correspondence of dynamics propagation and the disturbance is $A^{i-1}w_0 + A^{i-2}w_1 + \ldots + Aw_{i-2} + w_{i-1}$.

III. Robust Open-Loop MPC

Putting everything discussed above together, we can propose the **Robust Open-Loop MPC** scheme:

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

where $\mathcal{A}_i = [I \quad A \quad \ldots \quad A^{i-1}]$ is the dynamic propagation effect on the disturbance set and terminal set $\mathcal{X}_f$ is usually selected as the robust invariant set $\mathcal{O}_\infty^{\mathbb{W}}$ for some system $x^+ = (A+BK)x$ with stabilizing input gain K.

Again, the core idea of this robust open-loop MPC is: do nominal MPC, but with tighter constraints on the states and inputs. Then we can be sure that if the nominal system satisfies the tighter constraints, the uncertain system will satisfy the real constraints.

Note:

1. Robust open-loop MPC has a very small region of attraction, and in practice we would never use it. It is a theoretical development to give us the concepts of robust constraints satisfaction and robust MPC.
2. It can be shown robust open loop MPC has the property of robust control invariance: i.e., if $\mathbf{u}^*(x)$ is the optimizer of the robust open-loop MPC problem, then the system $Ax + Bu_0^*(x) + w \in \mathcal{X}$, for all $w \in \mathbb{W}$. This follows because the trajectory we computed at the current time is feasible for any disturbance, and therefore it's feasible for the one that we actually observe.
3. Also, by using the robust terminal set in robust open-loop MPC, we turn the robust control invariance problem into the vanilla robust invariance problem for autonomous system. This concept is analogous to what we have discussed before.

IV. Closed-Loop Predictions

As mentioned above, the robust open-loop MPC has a very small region of attraction. If the horizon is long, the feasible set after robust tightening would very likely be an empty set, and thus one can do nothing. But we have done nothing wrong in the settings above, which means our way to set up the problem has something off. Now we need to rethink how we can better establish the problem settings to deal with robust cases.

- **MPC as a Game**

Consider the system dynamics:

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

We can see that there are two players playing against each other: Controller vs. Disturbance:

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 (i.e. open-loop robust MPC)?

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

From the above analysis, we can see why robust open-loop MPC would always lead to a very limited region of attraction and feasible set. Because we are assuming that the **controller will do the same thing in the future no matter what the disturbance does!** This would largely benefit the disturbance and is unfavorable for the controller, because disturbance can always manipulate the dynamics in the worst case, but the controller can do nothing.

- **Closed-Loop Predictions**

Can we do better? The core idea is that **we need feedback in the loop!** Our prediction should be 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 (because $x_1 = Ax_0 + Bu_0 + w_0$)
4. Disturbance chooses his second move w_1 as a function of u_1
5. Controller decides his third move $u_1(x_2)$ **as a function of the first disturbance** w_0, w_1
6. **…**

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

Note:

1. Control policy is here is just another name for state **feedback** control law
2. Making μ a function of the state to input is the same as making μ a function of the disturbances from initial to time i to input since the state is a function of the disturbances up to that point.
3. The first input u_0 is a function of the current state. Therefore it is a single value, not a function.

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

Solution: Manually assume some specific structure on the functions μ_i. Following are some common choices used in different research literatures:

1. Pre-Stabilization: $\mu_i(x) = Kx + v_i$

   Fix K, such that $A+BK$ is stable Simple but often conservative
2. Linear Feedback: $\mu_i(x) = K_i x + v_i$

   Optimize over K_i and v_i. Nonconvex and usually extremely hard to solve
3. 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.
4. Tube MPC: $\mu_i(x) = K(x - \bar{x}_i) + v_i$

   Fix K, such that $A+BK$, optimize over $\bar{x}_i$ and v_i. Simple and can be quite effective

In this lecture, we would focus on the Tube MPC because it is the most practical among the above all.

Note: Recall that for LQR (linear, unconstrained, no disturbance) problem, or CLOC (linear, constrained, no disturbance) we also mentioned the so-called control policy (state feedback control law). For those simple cases, we proved that the policy has a specific structure: they are linear (LQR) or piece-wise affine (CLOC). But for robust case, we can only assume the structure and different assumptions are hard enough to be dug into as a Phd thesis topic!

V. Tube MPC

- **Overview**

Given a system with disturbance:

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

The core idea: Separate the available control authority into two parts

1. A portion that steers the noise-free (nominal) system to the origin $z^{+}=Az+Bv$
2. A portion that compensates for deviations from this system $e^{+}=(A+BK)e+w$ where the deviation (error) is $e=x-z$

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

Note: In other words, tube MPC has a hierarchical scheme. It divides the controller into two parts. On the top level there is a main MPC controller that works on the nominal system z, while at the bottom level there is a sub-controller that watches and compares the actual x and nominal z, and try to keep the deviation small to reduce the effect of disturbance (not necessary drive the deviation to zero). It can be regarded as a decoupling method to separately deal with the disturbance and nominal.

- **System Decomposition and Error Dynamics**

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:

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

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 and Minimal Invariant Set

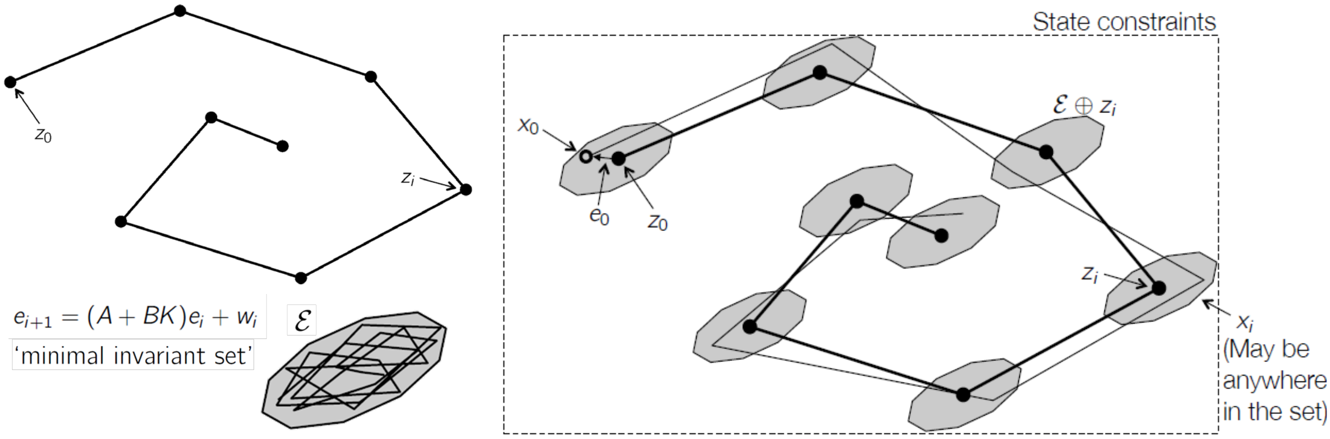
The error dynamics above, gives us the bounded maximum error, or how far the 'real' trajectory is from the nominal. Why? Consider

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

Here, **disturbance can be regarded as an input in error dynamics**. The autonomous dynamics part $(A+BK)$ is stable, and the input set $\mathbb{W}$ is bounded, so there is some set $\mathcal{E}$ that e will stay inside for all time. Making use of this, we want the smallest such set (the **'minimal invariant set'**)

As is shown below, we ignore the noise and plan the nominal trajectory, and we know that the real trajectory **stays nearby the nominal because we plan to apply the controller** $u_i = K(x_i - z_i) + v_i$ **in the future** (In other words, the trajectory would stay nearby because we have the stabilizing error dynamics, i.e. stabilizing K and thus have the minimal invariant set $\mathcal{E}$).

We then know that $x_i \in z_i \oplus \mathcal{E}$, and we want to ensure that all possible state trajectories satisfy the constraints ⇒ This is now equivalent to ensuring that $z_i \oplus \mathcal{E} \subseteq \mathcal{X}$



Note:

1. We won't actually do $u_i = K(x_i - z_i) + v_i$ in all the future because we would optimize over z_i and v_i, but $K(x_i - z_i) + v_i$ is a valid sub-optimal plan.
2. Satisfying input constraints is now more comple, which would be discussed later.
3. In the above figure, we can also see where the name "tube MPC" comes from. The nominal trajectory and the whole system state evolution are now enveloped by a "tube" whose "cross-section" is the minimal invariant set.

- **Computation and Problem Formulation**

After knowing the basic framework of the Tube MPC, we can conclude three things that need to be done, and we will discuss them one by one:

1. Compute the minimal invariant set $\mathcal{E}$ that the error will remain inside
2. Modify constraints on the nominal trajectory $\{z_i\}$
3. Formulate as a convex optimization problem and the final tube MPC

Uncertain State Evolution and Minimal Invariant Set Computation

Previously we wanted the maximum robust invariant set, or the largest set in which our terminal control law works. We now want the **minimum robust invariant set, or the smallest set that the state will remain inside despite the noise.**

Consider the system $x^+ = Ax + w$ and assume that $x_0 = 0$. We want to find out where the state can evolve to (i.e., how close can we stay to the origin):

$$
\begin{aligned}
x_1 &= w_0 \\
x_2 &= Ax_1 + w_1 = Aw_0 + w_1 \\
&\vdots \\
x_i &= \sum_{k=0}^{i-1} A^k w_k
\end{aligned}
$$

Assume that $w_i \in \mathbb{W}$ for all i. The set F_i hat contains all possible states x_i is

$$F_i = \bigoplus_{k=0}^{i-1} A^k \mathbb{W},\ F_0 = \{0\}$$

Where $P \oplus Q = \{x + y|\ x \in P,\ y \in Q\}$ is the Minkowski Sum.

If the system dynamics A is stable, as the sum goes to infinity, we arrive at (converge to) the minimum robust invariant set:

$$F_\infty = \bigoplus_{k=0}^{\infty} A^k \mathbb{W},\ F_0 = \{0\}$$

And if there exists an n such that $F_n = F_{n+1}$, then $F_n = F_\infty$, which leads to the following algorithm:

Algorithm (Algorithm to Compute Minimal Invariant Set):

Input: A, $\mathbb{W}$ (system dynamics, and disturbance set)

Output: F_∞ (minimal invariant set)

$\Omega_0 \leftarrow \{0\}$
loop
  $\Omega_{i+1} \leftarrow \Omega_i \oplus A^i \mathbb{W}$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $F_\infty = \Omega_i$
    end if
  end loop

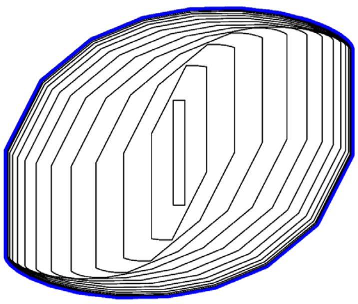
Note:

1. A finite n does not always exist, but large n can be a good but risky approximation (because the result is smaller than F_∞).
2. If n is not finite, we can also approximately compute the minimal invariant sets by taking an invariant outer approximation, which will be slightly larger than F_∞ but safe, and in practice we usually do this.
3. The Minkowski Sum of the polytopes is done by polytopic projection

Example: Minimal Invariant Set Computation

Given system dynamics where K is the LQR controller for $Q = I,\ R = 10$:

$$x^+ = \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} K\right)x + w,\ \ \mathbb{W} = \{w | |w_1| \le 0.01,\ |w_2| \le 0.1\}$$



Sets $A^i \mathbb{W}$ converging to F_∞ in the limit   The state trajectory will stay in F_∞ for all time

Note:

In the above discussion, we use a stable autonomous system whose state is x for general discussions, but always remember that we need to calculate the minimal robust invariant set on error dynamics described by $e_{i+1}=(A+BK)e_i+w_i,\ \ w_i\in\mathbb{W}$. Therefore, for calculating $\mathcal{E}$, we need to replace the above A to be $A_{\text{cl}}=(A+BK)$, and x_i is actually e_i

Modify and Tightened Constraints

Our goal is:

$$(x_i,u_i)\in\mathcal{X}\times\mathcal{U}\ \text{ for all }\ \{w_{0,\ldots},w_{i-1}\}\in\mathbb{W}^i$$

And now, we have decomposed the state x into the nominal one $z^+=Az+Bv$ and the deviation from the nominal (error) $e=x-z$, also, the input is composed of the nominal portion v and the error stabilization part: $u=K(x-z)+v$. Therefore, **the sufficient condition for constraints tightening** is:

Nominal state constraint tightening: $z_i\oplus\mathcal{E}\subseteq\mathcal{X}\ \Leftarrow\ z_i\in\mathcal{X}\ominus\mathcal{E}$

Input constraint tightening: $u_i\in K\mathcal{E}\oplus v_i\subseteq\mathcal{U}\ \Leftarrow\ v_i\in\mathcal{U}\ominus K\mathcal{E}$

Note:

1. The condition is only sufficient, not necessary and sufficient, because Minkowski sum is not the inverse operation of Pontryagin difference, recall the discussions in Lecture 9.
2. If the stabilizing gain K and the disturbance bounds are known in advance, The set $\mathcal{E}$ can be known offline and we can compute the tightened constraints in advance.

Tube-MPC Problem Formulation

Putting everything together, we formulate the tube MPC as the following convex optimization problem:

Feasible Set:
$$\mathcal{Z}(x_0)=\left\{\mathbf{z},\mathbf{v}\ \middle|\ \begin{array}{ll} z_{i+1}=Az_i+Bv_i & i=0,\ldots,N-1\\ z_i\in\mathcal{X}\ominus\mathcal{E} & i=0,\ldots,N-1\\ v_i\in\mathcal{U}\ominus K\mathcal{E} & i=0,\ldots,N-1\\ z_N\in\mathcal{X}_{f,\text{tube}} & \\ x_0\in z_0\oplus\mathcal{E} & \end{array}\right\}$$

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

Optimization Problem:
$$(\mathbf{z}^*(x_0),\mathbf{v}^*(x_0))=\underset{\mathbf{z},\mathbf{v}}{\operatorname{argmin}}\{V(\mathbf{z},\mathbf{v})\,|\,(\mathbf{z},\mathbf{v})\in\mathcal{Z}(x_0)\}$$

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

where $\mathbf{z}=\{z_0,\ldots,z_N\},\mathbf{v}=\{v_0,\ldots,v_{N-1}\}$ and here we use the subscript tube $\mathcal{X}_{f,\text{tube}}$ to represent the tightened terminal set for the nominal system (Terminal set notation is slightly different from the slides to distinguish from $\mathcal{X}_f$ we used before, see the notes later.)

- **Feasibility and Stability of tube MPC**

In this section, we would discuss the feasibility and stability of tube MPC. Actually, it is much the same as for nominal MPC, we need three assupmtions on the system for theoretical proof:

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 for the nominal system under the local control law $\kappa_f(z)$:

$$z^+ = Az + B\kappa_f(z) \in \mathcal{X}_{f,\text{tube}} \quad \text{for all} \quad z \in \mathcal{X}_{f,\text{tube}}$$

   and all tightened state and input constraints are satisfied in $\mathcal{X}_{f,\text{tube}}$

$$\mathcal{X}_f \subseteq \mathcal{X} \ominus \mathcal{E},\ \kappa_f(z) \in \mathcal{U} \ominus K\mathcal{E} \quad \text{for all} \quad z \in \mathcal{X}_{f,\text{tube}}$$

3. Terminal cost is a continuous Lyapunov function in the terminal set:

$$V_f(Az + B\kappa_f(z)) - V_f(z) \leq l(z, \kappa_f(z)) \quad \text{for all} \quad z \in \mathcal{X}_{f,\text{tube}}$$

With these three assumptions, we can prove the robust recursive feasibility and stability of tube MPC:

Robust Feasibility (Invariance):

Let $\mathbf{z}^* = \{z_0^*, \ldots, z_N^*\}, \mathbf{v}^* = \{v_0^*, \ldots, v_{N-1}^*\}$ be the optimal solution of tube MPC problem for time x_0

At the next timestep, the state is:

$$x_1 = Ax_0 + BK(x_0 - z_0^*) + Bv_0^* + w \quad \text{for some} \quad w \in \mathbb{W}$$

i.e., the state x_1 may have many possible values. We need to show that there exists a feasible solution **for all of them**

By construction, the state x_1 is in the set $z_1 \oplus \mathcal{E}$ for all $\mathbb{W}$. Therefore (as in standard MPC), the sequence

$$(\mathbf{z}^*, \mathbf{v}^*) = (\{z_1^*, \ldots, z_N^*, Az_N^* + B\kappa_f(z_N^*)\}, \{v_1^*, \ldots, v_{N-1}^*, \kappa_f(z_N^*)\})$$

is feasible for all x_1

That is to say, set $\{x |\ \mathcal{Z}(x) \neq \emptyset\}$ is a robust invariant set of the system $x^+ = Ax + B\mu_{\text{tube}}(x) + w$ subject to the constraints $(x,u) \in \mathcal{X} \times \mathcal{U}$ where $\mathcal{Z}(x)$ is the feasible set defined for the tube MPC problem shown above.

Robust Stability:

As in standard MPC, we have the relationship:

$$J^*(x_0) = \sum_{i=0}^{N-1} l(z_i^*, v_i^*) + V_f(z_N^*)$$

$$J^*(x_1) \leq \sum_{i=1}^{N-1} l(z_i^*, v_i^*) + V_f(z_{N+1}^*)$$

$$= J^*(x_0) - \underbrace{l(z_0^*, v_0^*)}_{\geq 0} + \underbrace{V_f(z_{N-1}^*) - V_f(z_N) + l(z_N^*, \kappa_f(z_N^*))}_{\leq 0,\ \because V_f(Az + B\kappa_f(z)) - V_f(z) \leq l(z, \kappa_f(z)),\ \forall z \in \mathcal{X}_{f,\text{tube}}}$$

This shows that $\lim_{i\to\infty} J(z_0^*(x_i)) = 0$, and therefore $\lim_{i\to\infty} z_0^*(x_i) = 0$. However, note that z is for the nominal system, the actual state x_i does not tend to zero! It only stays within a robust invariant set centered at $z_0^*(x_i) : \lim_{i\to\infty} \text{dist}(x_i, \mathcal{E}) = 0$, where dist is any distance function.

In other words, we can say that the state x of the system $x^+ = Ax + B\mu_{\text{tube}}(x) + w$ converges in the limit to the set $\mathcal{E}$.

- **Design Pipeline, Example and Summary**

Putting all the things together, we now show how to implement tube MPC and give an example

Offline Phase

1. Choose a stabilizing controller K such that $|A+BK|<1$
2. Compute the minimal robust invariant set $\mathcal{E}=F_{\infty}$ for the error dynamics $e^{+}=(A+BK)e+w$, $w\in\mathbb{W}^{1}$
3. Compute the tightened constraints for the nominal system $\mathcal{X}\ominus\mathcal{E}$, $\mathcal{U}\ominus K\mathcal{E}$
4. Choose a terminal weight function V_f and terminal set $\mathcal{X}_{f,\text{tube}}$ satisfying the three assumptions discussed above.

Online Phase

1. Measure / estimate state x
2. Solve the problem $(\mathbf{z}^{*}(x_0),\mathbf{v}^{*}(x_0))=\underset{\mathbf{z},\mathbf{v}}{\operatorname{argmin}}\{V(\mathbf{z},\mathbf{v})\,|\,(\mathbf{z},\mathbf{v})\in\mathcal{Z}(x_0)\}$
3. Set the input to $u=\mu_{\text{tube}}(x)=K(x-z_0^{*}(x))+v_0^{*}(x)$

Example: Tube MPC for 2D system

System dynamics:

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

Constraints:

$$\mathcal{X}=\{x\,|\,\|x\|_{\infty}\le 1\},\ \mathcal{U}\{u\,|\,\|u\|_{\infty}\le 1\}$$

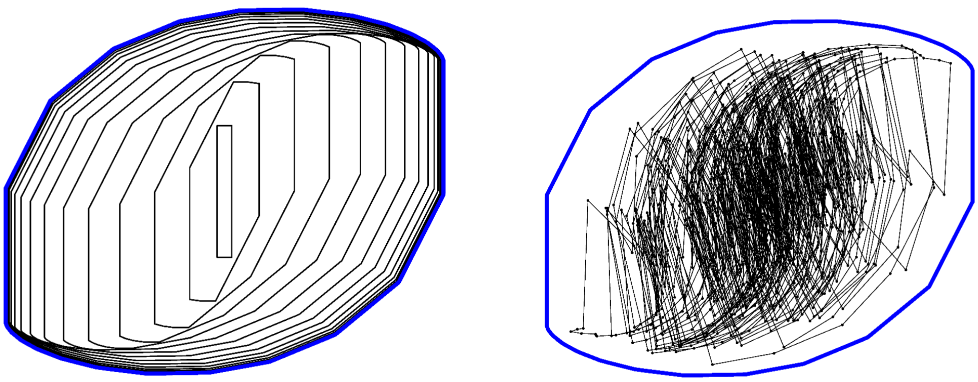
Stage cost is:

$$l(z,v)=z_i^{\mathrm{T}}Qz_i+v_i^{\mathrm{T}}Rv_i\ \text{ where }\ Q=I,\ R=10$$

1. Choose a stabilizing controller K such that $|A+BK|<1$

   We take the LQR controller for $Q=I,\ R=10\Rightarrow K=[-0.5198\ \ -0.9400]$
2. Compute the minimal robust invariant set $\mathcal{E}=F_{\infty}$ for the error dynamics $e^{+}=(A+BK)e+w$

   As the following blue region shows:



3. Compute the tightened constraints for the nominal system, let $\mathcal{E}=\{e|\ Fe\le f\}$, we have:

$$\mathcal{X}=\{x\,|\,\|x\|_{\infty}\le 1\}=\left\{x\left|\begin{bmatrix}I\\-I\end{bmatrix}x\le\begin{bmatrix}\mathbf{1}\\\mathbf{1}\end{bmatrix}\right.\right\}$$

$$\mathcal{X} \ominus \mathcal{E} = \{x \mid x + e \in \mathcal{X},\ \forall e \in \mathcal{E}\} = \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x + \begin{bmatrix} I \\ -I \end{bmatrix} e \le \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix},\ \forall e \in \mathcal{E} \right\}$$

$$= \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x \le \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix} e,\ \forall e \in \mathcal{E} \right\}$$

$$= \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x \le \min \left\{ \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix} e \right\},\ e \in \mathcal{E} \right\}$$

$$= \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x \le \begin{bmatrix} 1 + \min\{[-1 \quad 0]e\} \\ 1 + \min\{[0 \quad -1]e\} \\ 1 + \min\{[1 \quad 0]e\} \\ 1 + \min\{[0 \quad 1]e\} \end{bmatrix} \,\middle|\, Fe \le f \right\}$$

Similarly, for input tightening we have:

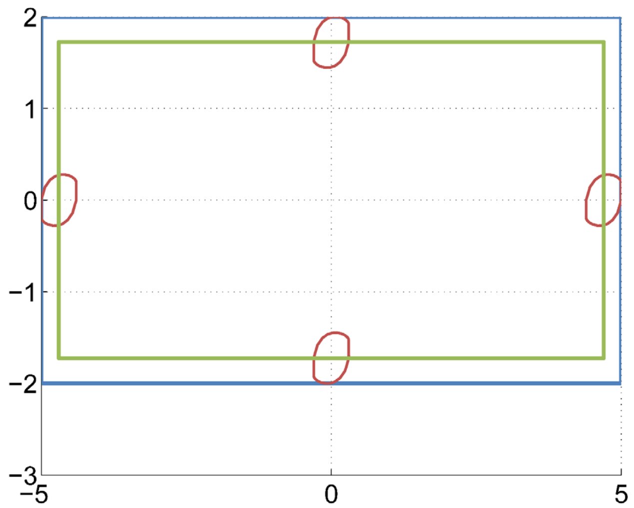
$$\mathcal{U} \ominus K\mathcal{E} = \left\{ u \,\middle|\, \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \ominus \{Ke \mid e \in \mathcal{E}\}$$

$$= \left\{ u \,\middle|\, \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} Ke,\ \forall e \in \mathcal{E} \right\}$$

$$= \left\{ u \,\middle|\, \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le \min \left\{ \begin{bmatrix} 1 - Ke \\ 1 + Ke \end{bmatrix} \right\} \,\middle|\, Fe \le f \right\}$$

$$= \left\{ u \,\middle|\, \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le \begin{bmatrix} 1 + \min\{-Ke\} \\ 1 + \min\{Ke\} \end{bmatrix} \,\middle|\, Fe \le f \right\}$$

An illustration of state constraint tightening is shown below:



Blue : Original constraint set $\mathcal{X}$

Red : Minimal robust invariant error set $\mathcal{E}$

Green : Tightened constraints $\mathcal{X} \ominus \mathcal{E}$

4. Choose the terminal weight function V_f and terminal set $\mathcal{X}_{f,\text{tube}}$
   Choose the terminal control law to the the LQR control law $\kappa_f(z) = Kz$ where the weights Q and R are taken the same as for our MPC problem. Computue maximal invariant set $\mathcal{O}_{\infty,\text{tube}}$ for the autonomous nominal linear system $z^+ = (A + BK)z = A_{\text{cl}} z$ with state and input constraints $x \in \mathcal{X} \ominus \mathcal{E}$, $Kx \in \mathcal{U} \ominus K\mathcal{E}$ (recall previous lectures).

Set $\mathcal{X}_{f,\text{tube}} = \mathcal{O}_{\infty,\text{tube}}$, so that $\mathcal{X}_{f,\text{tube}}$ would satisfy:

$$\text{pre}(\mathcal{X}_{f,\text{tube}}) = \mathcal{X}_{f,\text{tube}},\ \mathcal{X}_{f,\text{tube}} \subseteq \mathcal{X} \ominus \mathcal{E},\ K\mathcal{X}_{f,\text{tube}} \subseteq \mathcal{U} \ominus K\mathcal{E}$$

And we need to find a function V_f satisfying the property we need. Recall that the optimal cost of the LQR control law is:

$$V^*(z_0) = \sum_{i=0}^{\infty} z_i^{\mathrm{T}}(Q + K^{\mathrm{T}}RK)z_i = z_0^{\mathrm{T}} P z_0$$

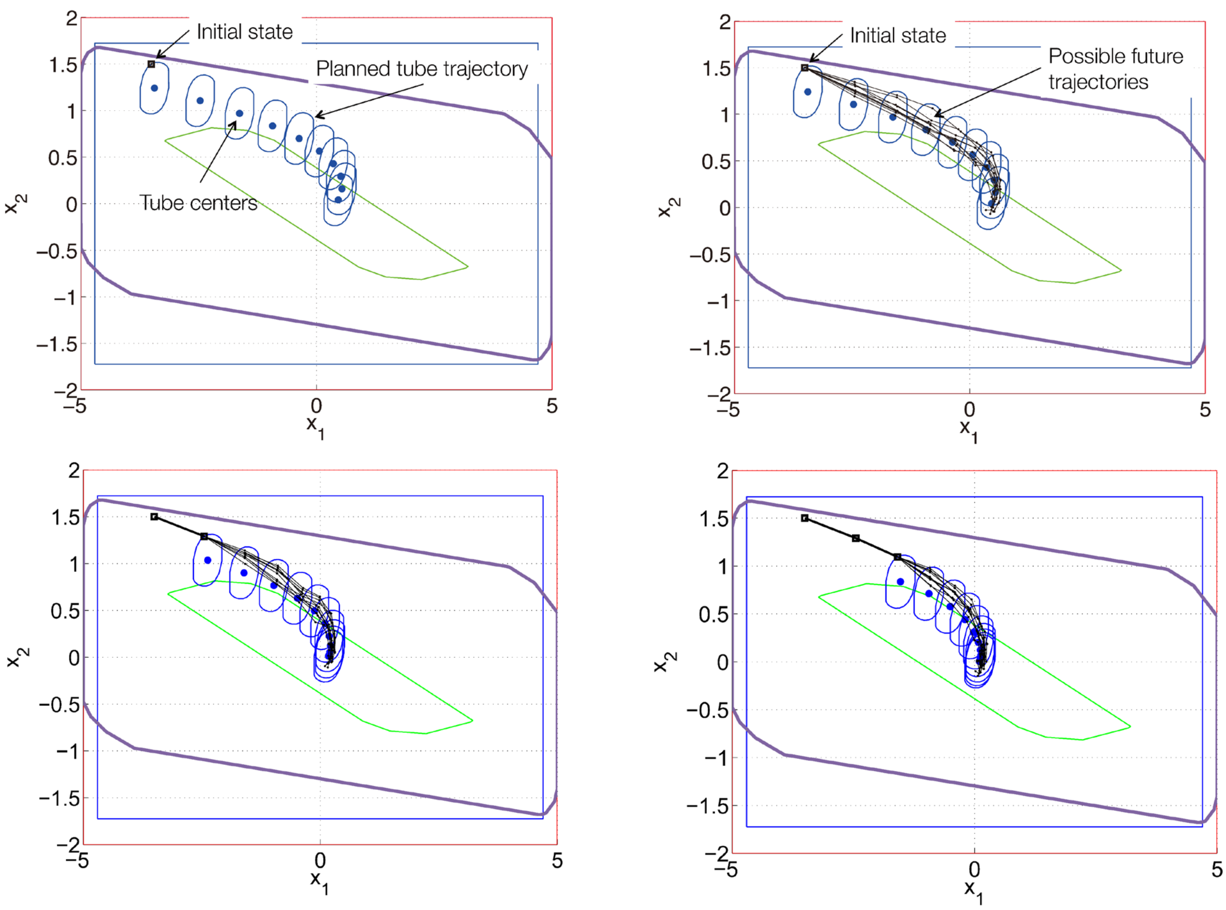
where P is the solution to a discrete-time Riccati equation, and we know that $V^*(z)$ is a Lyapunov function for system $z^+ = (A + BK)z$:

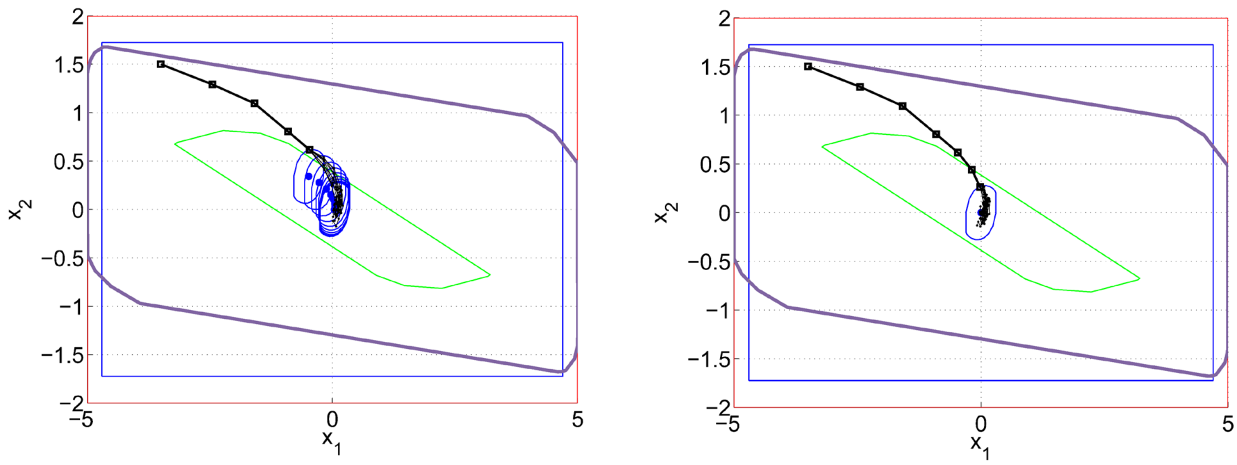
$$V^*(z_1) - V^*(z_0) = \sum_{i=1}^{\infty} z_i^{\mathrm{T}}(Q + K^{\mathrm{T}}RK)z_i - \sum_{i=0}^{\infty} z_i^{\mathrm{T}}(Q + K^{\mathrm{T}}RK)z_i$$
$$= -z_0^{\mathrm{T}}(Q + K^{\mathrm{T}}RK)z_0 = -l(z_0, \kappa_f(z_0))$$

which is exactly what we need, and therefore, we can take $V_f(z) = z^{\mathrm{T}} P z$

Note: we can see that the terminal set calculation is different from vanilla MPC. In tube MPC, we need to set $\mathcal{X}_{f,\text{tube}}$ to be the invariant set calculated by taking the state and input constraints to be $\mathcal{X} \ominus \mathcal{E}$, $\mathcal{U} \ominus K\mathcal{E}$ for the nominal system. So that the resulting terminal set can satisfy the tightened constraints and thus the three assumptions we needed.

The online execution result of tube MPC is shown below:





Summary

All in all, tube MPC can be summarized as follows:

Idea:

Split input into two parts: One to steer the 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). Even works for unstable systems, and the optimization problem to solve is simple.

Cons:

Sub-optimal MPC (optimal is extremely difficult, but actually, we would not care that much about the optimality in robust cases since the cost is also defined on some nominal trajectory)

Reduced feasible set compared to nominal MPC (this is the common problem for robust controllers)

We need to know what $\mathbb{W}$ is (this is usually not realistic)