Model Predictive Control

Lecture: Robust MPC

Colin Jones

Laboratoire d'Automatique, EPFL

Recap: Three Key Issues that are Always Present

Feasible Set

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

$\Rightarrow$ Drop terminal set

$\Rightarrow$ Soften constraints

Tracking

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

$\Rightarrow$ Move origin and solve regulation problem

Disturbance rejection

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

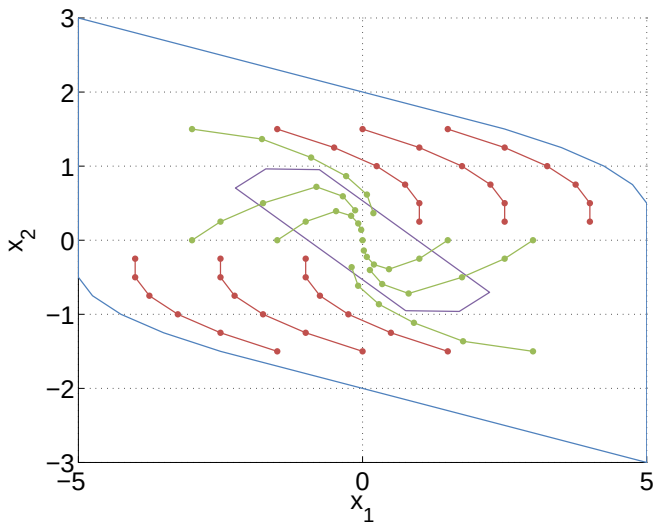
$\Rightarrow$ Estimate disturbance and compensate by 'tracking' to artificial target

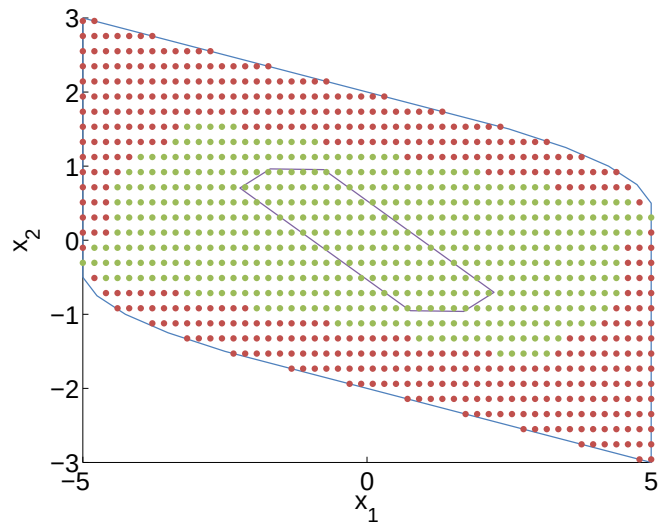
MPC without Terminal Set

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 constraint is satisfied without enforcing it
$\Rightarrow$ Solution of finite horizon MPC problem = infinite horizon solution





Downsides:

- Loose recursive feasibility $\rightarrow$ Feasible now, does not mean feasible later
- Characterizing invariant region extremely difficult $\rightarrow$ It may work, it may not

Soft constrained MPC problem setup

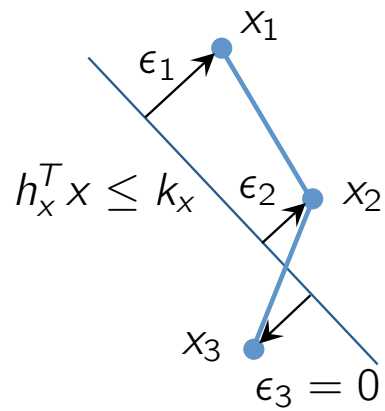
$$\min_{\mathbf{u}} \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + \rho(\epsilon_i) + x_N^T P x_N$$

$$\text{s.t. } x_{i+1} = Ax_i + Bu_i$$

$$H_x x_i \leq k_x + \epsilon_i,$$

$$H_u u_i \leq k_u,$$

$$\epsilon_i \geq 0$$



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

Pros/Cons

- Problem is always feasible (critical!)
- Tune ρ for tradeoff between amount of violation and duration (difficult)
- No theory: May not be stabilizing

$\Rightarrow$ Always used in practice

Recap: Three Key Issues that are Always Present

Feasible Set

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

$\Rightarrow$ Drop terminal set

$\Rightarrow$ Soften constraints

Tracking

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

$\Rightarrow$ Move origin and solve regulation problem

Disturbance rejection

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

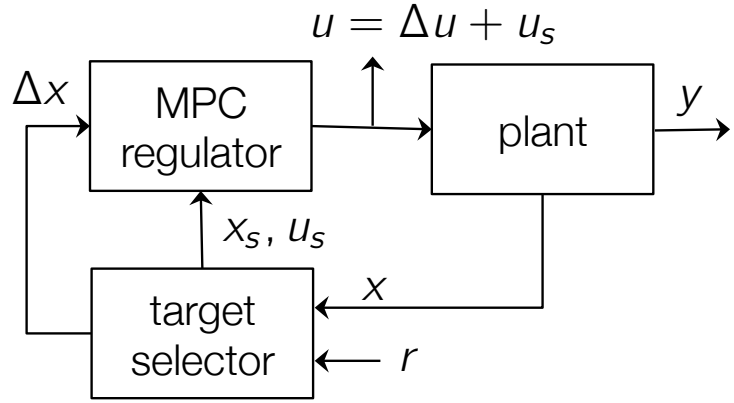
$\Rightarrow$ Estimate disturbance and compensate by 'tracking' to artificial target

MPC problem for tracking

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

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

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



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

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:

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

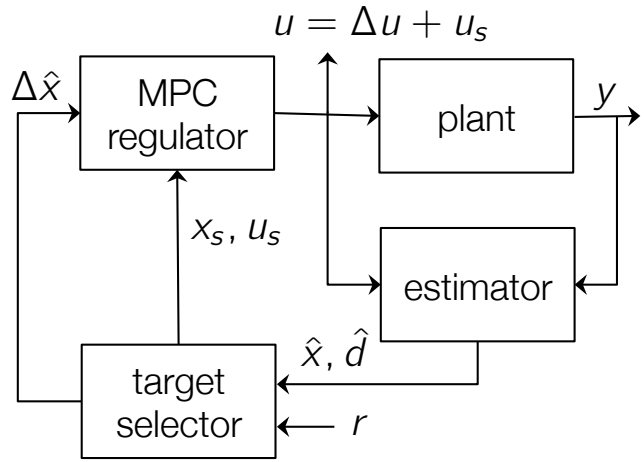
$$\text{s.t. } \Delta x_0 = \Delta\hat{x}$$

$$\Delta x_{i+1} = A\Delta x_i + B\Delta u_i$$

$$H_x \Delta x_i \leq k_x - H_x x_s$$

$$H_u \Delta u_i \leq k_u - H_u u_s$$

$$\Delta x_N \in \mathcal{X}_f$$



Lecture Take Homes

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

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

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

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

Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise
   - Choosing a cost to minimize
   - Robust Constraint Satisfaction

3. Robust Open-Loop MPC

MPC vs The Real World

Predictive control assumption:

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

- System evolves in a predictable fashion

The real world:

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

- Random noise w changes the evolution of the system
- Model structure is unknown
- Unknown parameters θ impact the dynamics

(w changes with time, θ is unknown, but constant)

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

Recall: Goals of Constrained Control

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

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

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

What if f is only known approximately?

Goals of Robust Constrained Control

Uncertain Constrained System
$x^+ = f(x, u, w; \theta) \qquad (x, u) \in \mathbb{X}, \mathbb{U} \quad w \in \mathbb{W} \quad \theta \in \Theta$

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

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

Meeting these goals requires some knowledge/assumptions about the random values w and θ.

Examples of Common Uncertainty Models

Measurement / Input Bias
$$g(x, u, w; \theta) = f(x, u) + \theta$$
θ unknown, but constant

- Unexpected offset can cause constraint violation
- Offset doesn't change, or changes slowly with time
  - $\rightarrow$ Generally handled by estimating offset and compensating (last week's lecture)
- Constraint violation still possible before offset is estimated

Examples of Common Uncertainty Models

Linear Parameter Varying System

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

A_k, B_k known, θ_k unknown, but constant

- Actual system is linear - but exact dynamics unknown
- Preventing constraint violation requires considering **all** possible trajectories (very conservative)
- Often handled by estimating θ (adaptive control), since it is constant, or changes slowly
  - Very difficult if system is unstable

Examples of Common Uncertainty Models

Polytopic Uncertainty

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

A_k, B_k known, w_k unknown and changing at each sample time

- Dynamics change randomly at each point in time $\rightarrow$ nonlinear system
- Preventing constraint violation requires considering **all** possible trajectories (not conservative, since they can all happen)
- Commonly dealt with via **robust MPC**

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

Examples of Common Uncertainty Models

Additive Stochastic Noise
$$g(x, u, w; \theta) = Ax + Bu + w$$
Distribution of w known

- Distribution of the disturbance is known
- Problem significantly more challenging (even to formulate the goals)
- Topic of active research

Examples of Common Uncertainty Models

Additive Bounded Noise
$$g(x, u, w; \theta) = Ax + Bu + w \ , \quad w \in \mathbb{W}$$
A, B known, w unknown and changing with each sample

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

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

Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise
   - Choosing a cost to minimize
   - Robust Constraint Satisfaction

3. Robust Open-Loop MPC

Goals of Robust Constrained Control

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

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

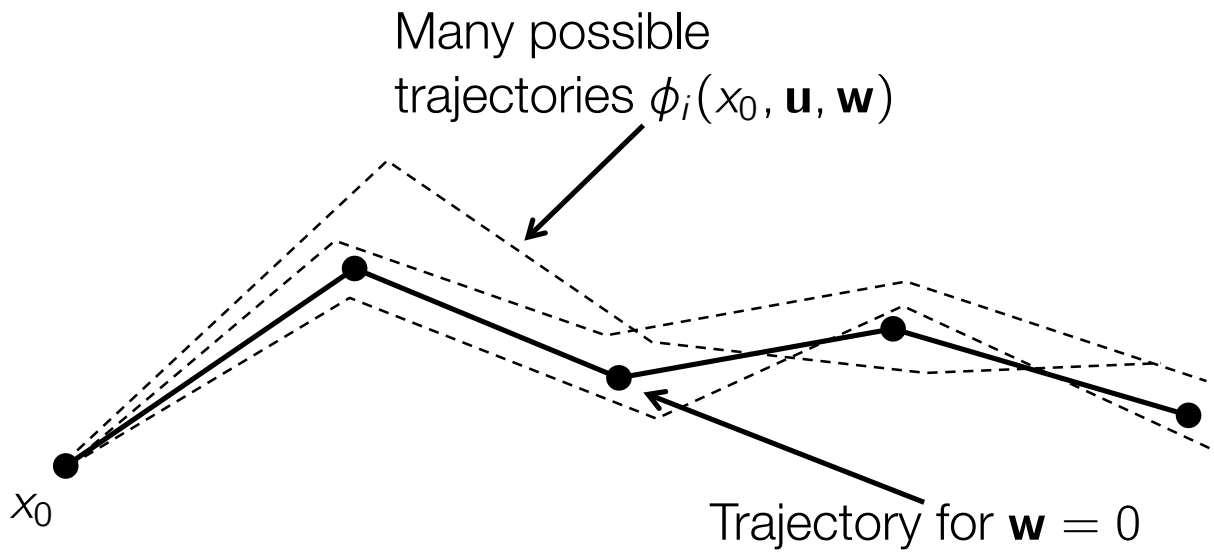
1. Satifies constraints : $\{x_i\} \subset \mathbb{X}$, $\{u_i\} \subset \mathbb{U}$ for all disturbance realizations
2. Is stable: Converges to a neighbourhood of the origin
3. Optimizes (expected/worst-case) "performance"
4. Maximizes the set $\{x_0 \mid \text{Conditions 1-3 are met}\}$

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

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

Uncertain State Evolution

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



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

Uncertain State Evolution

Nominal system

$$x^+ = Ax + Bu$$

$$x_1 = Ax_0 + Bu_0$$
$$x_2 = A^2x_0 + ABu_0 + Bu_1$$
$$\vdots$$
$$x_i = A^ix_0 + \sum_{k=0}^{i-1} ABu_{i-k}$$

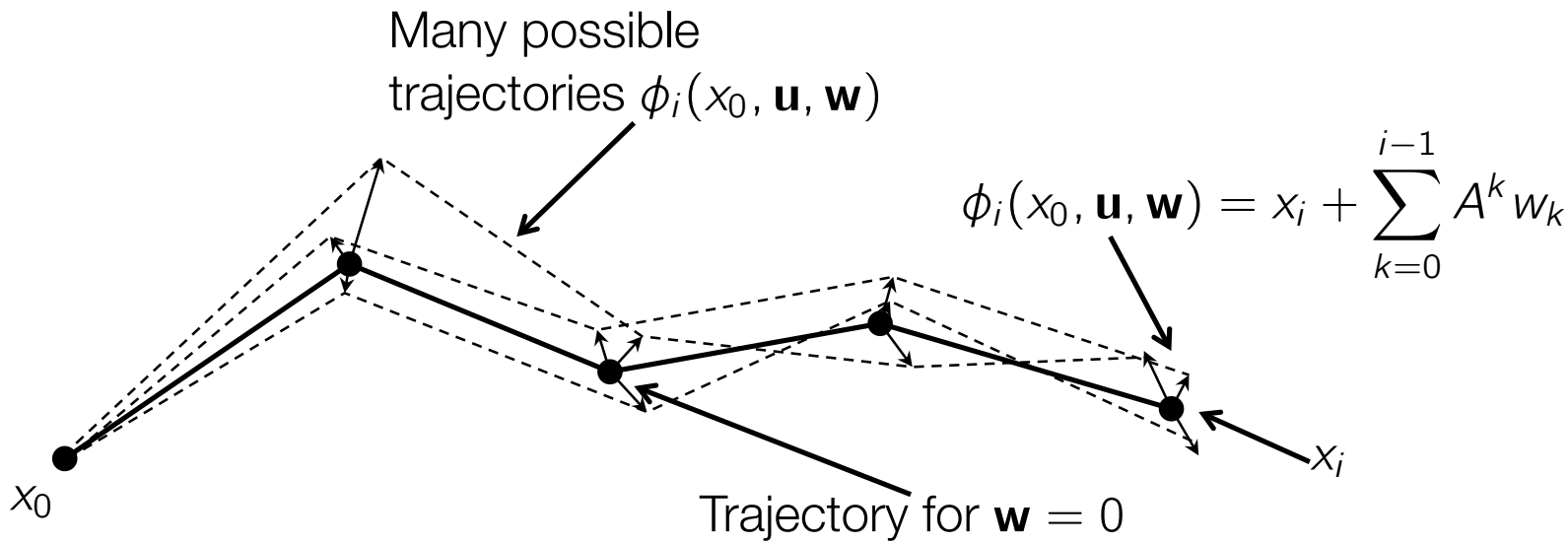
Uncertain system

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

$$\phi_1 = Ax_0 + Bu_0 + w_0$$
$$\phi_2 = A^2x_0 + ABu_0 + Bu_1 + Aw_0 + w_1$$
$$\vdots$$
$$\phi_i = A^ix_0 + \sum_{k=0}^{i-1} A^kBu_{i-k} + \sum_{k=0}^{i-1} A^kw_{i-k}$$
$$\phi_i = x_i + \sum_{k=0}^{i-1} A^kw_{i-k}$$

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

Uncertain State Evolution



Goals of Robust Constrained Control

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

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

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

Defining a Cost to Minimize

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

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

However, there are now many trajectories that *may* occur, depending on the disturbance $\mathbf{w}$.

The cost is now a function of the disturbance seen, and therefore *each possible trajectory has a different cost*:

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

Need to 'eliminate' the dependence on $\mathbf{w}$.

Defining a Cost to Minimize

Several common options:

- Minimize the expected value (requires some assumption on the distribution)

$$V_N(x_0, \mathbf{u}) := \mathbf{E}\left[J(x_0, \mathbf{u}, \mathbf{w})\right]$$

- Minimize the variance (requires some assumption on the distribution)

$$V_N(x_0, \mathbf{u}) := \mathrm{Var}\left(J(x_0, \mathbf{u}, \mathbf{w})\right)$$

- Take the worst-case

$$V_N(x_0, \mathbf{u}) := \max_{\mathbf{w} \in \mathbb{W}^{N-1}} J(x_0, \mathbf{u}, \mathbf{w})$$

- Take the nominal case

$$V_N(x_0, \mathbf{u}) := J(x_0, \mathbf{u}, 0)$$

Defining a Cost to Minimize

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

$$V_N(x_0, \mathbf{u}) := J(x_0, \mathbf{u}, 0)$$

- We will 'fluff' over the stability proof, because we cannot demonstrate robust stability in this case (i.e., asymptotic convergence for all possible disturbances).
- The next lecture will introduce a new notion of stability that will allow us to analyse this case

Goals of Robust Constrained Control

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

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

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

Robust Constraint Satisfaction

Recall: We break the MPC prediction into two parts.

$$\left.\begin{aligned} \phi_{i+1} &= A\phi_i + Bu_i + w_i \\ u_i &\in \mathbb{U} \\ \phi_i &\in \mathbb{X}\ \forall \mathbf{w} \in \mathbb{W}^N \end{aligned}\right\}$$

- $i = 0, \ldots, N-1$
- Optimize over control actions $\{u_0, \ldots, u_{N-1}\}$
- Enforce constraints explicitly by imposing $\phi_i \in \mathbb{X}$ and $u_i \in \mathbb{U}$ for all sequences $\mathbf{w}$

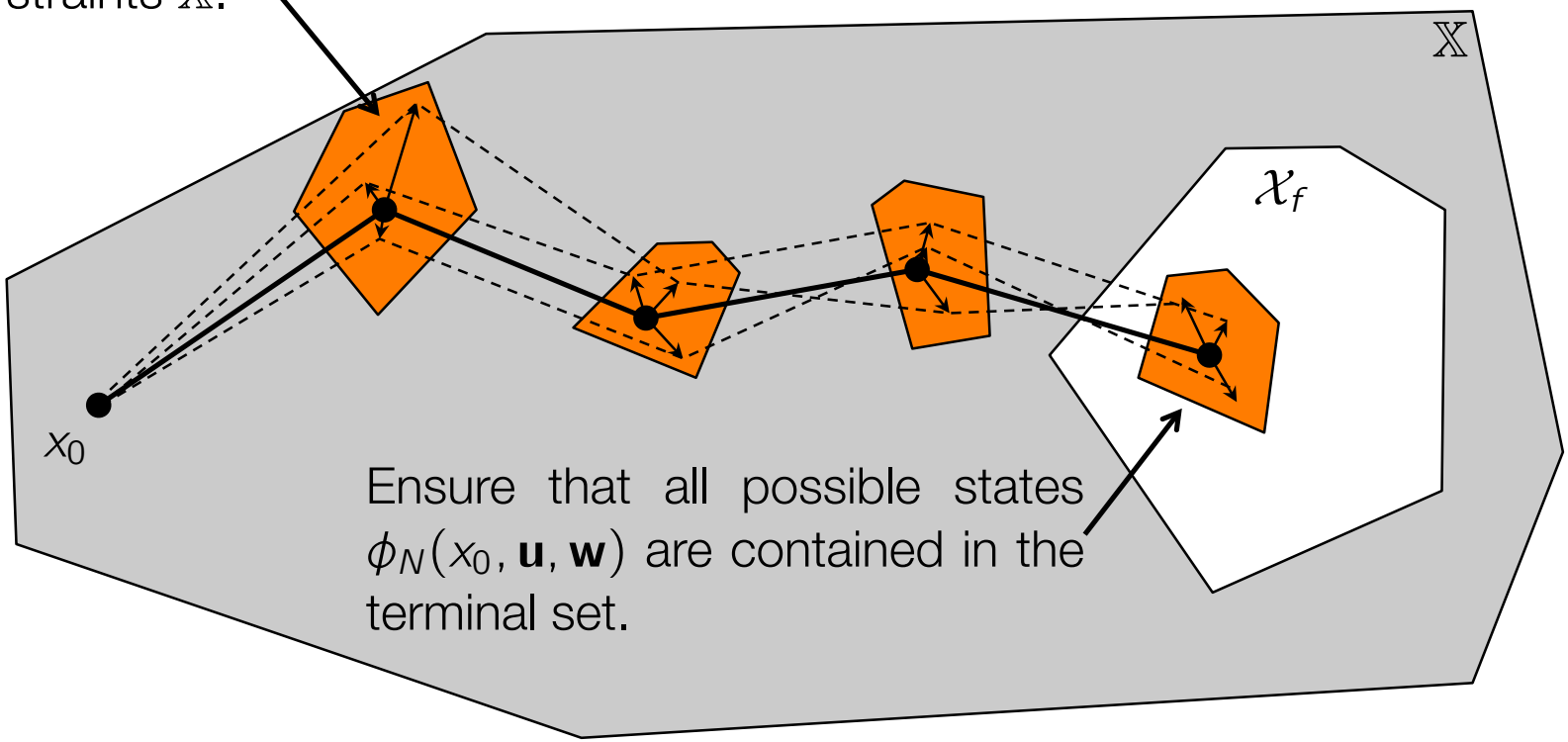
$$\left.\begin{aligned} \phi_{i+1} &= (A + BK)\phi_i + w_i \\ \phi_N &\in \mathcal{X}_f \end{aligned}\right\}$$

- $i = N, \ldots$
- Assume control law to be linear $u_i = K\phi_i$
- Enforce constraints implicitly by constraining ϕ_N to be in an *robust invariant set* $\mathcal{X}_f \subseteq \mathbb{X}$ and $K\mathcal{X}_f \subseteq \mathbb{U}$ for the system $\phi^+ = (A + BK)\phi + w$

In the following:

- Robustly enforcing constraints of a linear system
- Robustly ensuring constraints of the sequence $\phi_1, \ldots, \phi_{N-1}$

Robust Constraint Satisfaction



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

Robust Constraint Satisfaction

Recall: We break the MPC prediction into two parts.

$$\left.\begin{aligned} \phi_{i+1} &= A\phi_i + Bu_i + w_i \\ u_i &\in \mathbb{U} \\ \phi_i &\in \mathbb{X}\ \forall \mathbf{w} \in \mathbb{W}^N \end{aligned}\right\}$$

- $i = 0, \ldots, N-1$
- Optimize over control actions $\{u_0, \ldots, u_{N-1}\}$
- Enforce constraints explicitly by imposing $\phi_i \in \mathbb{X}$ and $u_i \in \mathbb{U}$ for all sequences $\mathbf{w}$

$$\left.\begin{aligned} \phi_{i+1} &= (A + BK)\phi_i + w_i \\ \phi_N &\in \mathcal{X}_f \end{aligned}\right\}$$

- $i = N, \ldots$
- Assume control law to be linear $u_i = K\phi_i$
- Enforce constraints implicitly by constraining ϕ_N to be in an *robust invariant set* $\mathcal{X}_f \subseteq \mathbb{X}$ and $K\mathcal{X}_f \subseteq \mathbb{U}$ for the system $\phi^+ = (A + BK)\phi + w$

In the following:

- **Robustly enforcing constraints of a linear system**
- Robustly ensuring constraints of the sequence $\phi_1, \ldots, \phi_{N-1}$

Reminder: Invariance

Constraint satisfaction, for an **autonomous** system $x^+ = f(x)$, or **closed-loop** system $x^+ = f(x, \kappa(x))$ for a **given** controller κ.

Positive Invariant set
A set $\mathcal{O}$ is said to be a positive invariant set for the autonomous system $x_{i+1} = f(x_i)$ if $$x_i \in \mathcal{O} \;\Rightarrow\; x_i \in \mathcal{O} \;,\; \forall i \in \{0, 1, \dots\}$$

If we have an invariant set $\mathcal{X}_f \subseteq \mathbb{X}$ and $\kappa(\mathcal{X}_f) \subseteq \mathbb{U}$, then it provides a set of initial states from which the trajectory will never violate the system constraints if we apply the controller κ.

Robust Invariant Set

Robust constraint satisfaction, for an **autonomous** system $x^+ = f(x, w)$, or **closed-loop** system $x^+ = f(x, \kappa(x), w)$ for a **given** controller κ.

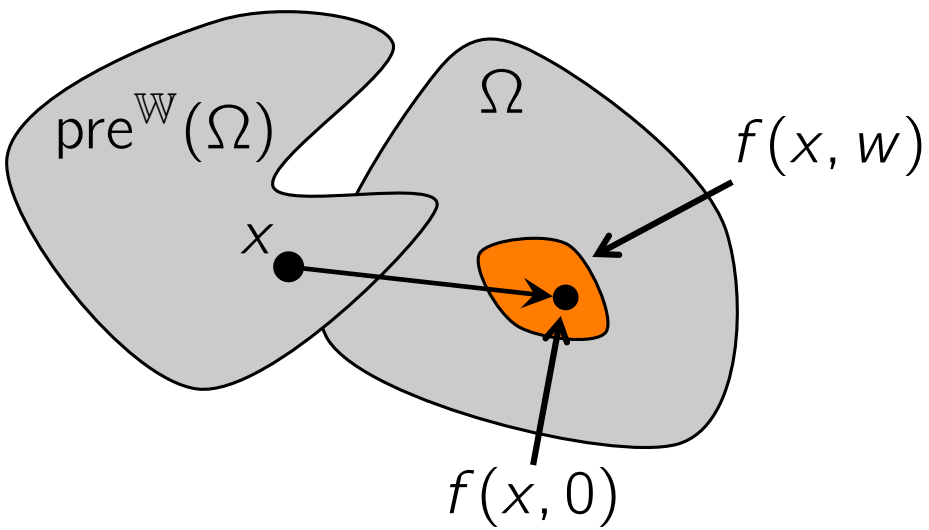
Robust Positive Invariant set
A set $\mathcal{O}^{\mathbb{W}}$ is said to be a robust positive invariant set for the autonomous system $x_{i+1} = f(x_i, w)$ if $$x \in \mathcal{O}^{\mathbb{W}} \;\Rightarrow\; f(x, w) \in \mathcal{O}^{\mathbb{W}} \;, \text{ for all } w \in \mathbb{W}$$

Robust Pre-Sets

Robust Pre Set

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

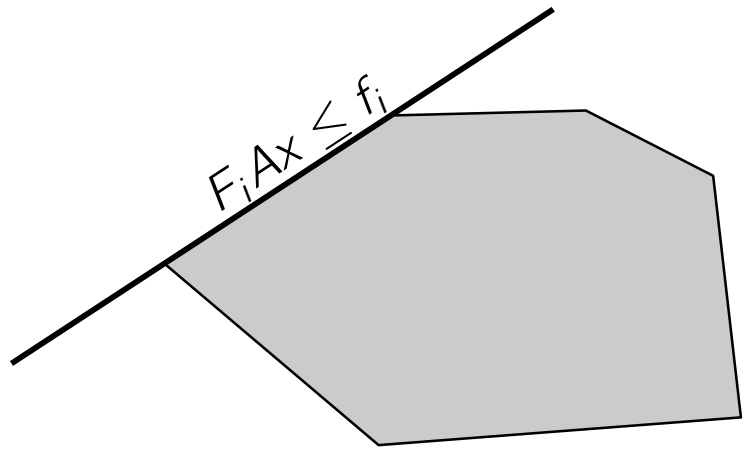
$$\text{pre}^{\mathbb{W}}(\Omega) := \{x \mid f(x, w) \in \Omega \text{ for all } w \in \mathbb{W}\}$$



Computing Robust Pre-Sets for Linear Systems

Goal: Given the system $f(x, w) = Ax + w$, and the set $\Omega := \{x \mid Fx \leq f\}$, compute $\text{pre}^{\mathbb{W}}(\Omega)$.

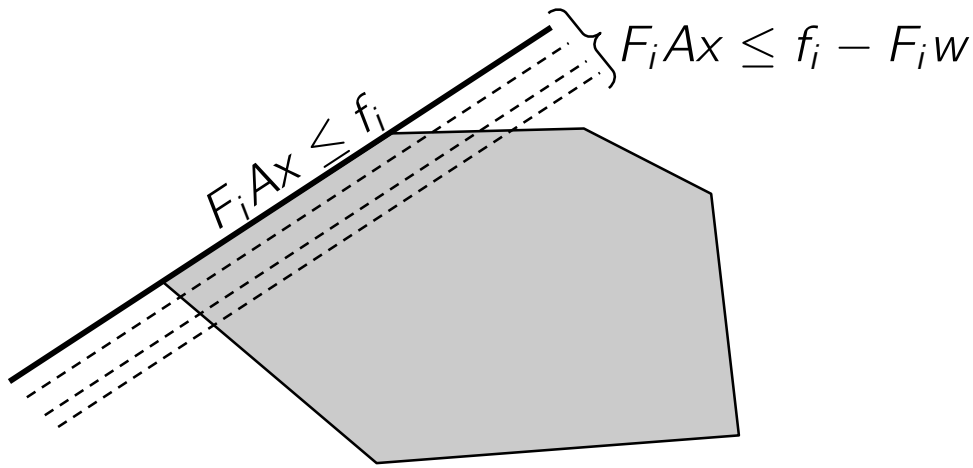
$$\text{pre}^{\mathbb{W}}(\Omega) = \{x \mid Ax + w \in \Omega,\ \forall w \in \mathbb{W}\} = \{x \mid FAx + Fw \leq f,\ \forall w \in \mathbb{W}\}$$



Computing Robust Pre-Sets for Linear Systems

Goal: Given the system $f(x, w) = Ax + w$, and the set $\Omega := \{x \mid Fx \leq f\}$, compute $\text{pre}^{\mathbb{W}}(\Omega)$.

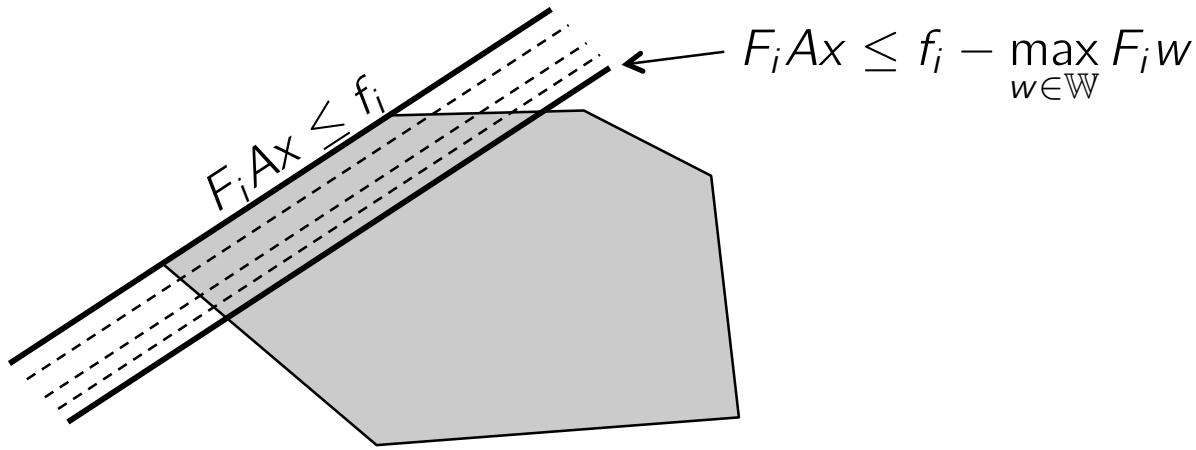
$$\text{pre}^{\mathbb{W}}(\Omega) = \{x \mid Ax + w \in \Omega,\ \forall w \in \mathbb{W}\} = \{x \mid FAx + Fw \leq f,\ \forall w \in \mathbb{W}\}$$



Computing Robust Pre-Sets for Linear Systems

Goal: Given the system $f(x, w) = Ax + w$, and the set $\Omega := \{x \mid Fx \leq f\}$, compute $\text{pre}^{\mathbb{W}}(\Omega)$.

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

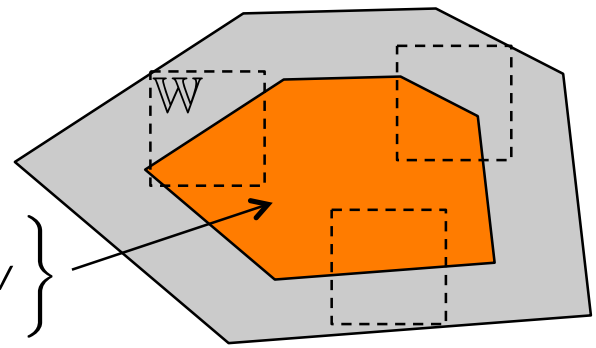


Computing Robust Pre-Sets for Linear Systems

Goal: Given the system $f(x, w) = Ax + w$, and the set $\Omega := \{x \mid Fx \leq f\}$, compute $\text{pre}^{\mathbb{W}}(\Omega)$.

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

$$\left\{ x \;\middle|\; FAx \leq f - \max_{w \in \mathbb{W}} Fw \right\}$$



$$\text{pre}^{\mathbb{W}}(\Omega) = \left\{ x \;\middle|\; FAx \leq f - \max_{w \in \mathbb{W}} Fw \right\} = \{x \mid FAx \leq f - h_{\mathbb{W}}(F)\} = A(\Omega \ominus \mathbb{W})$$

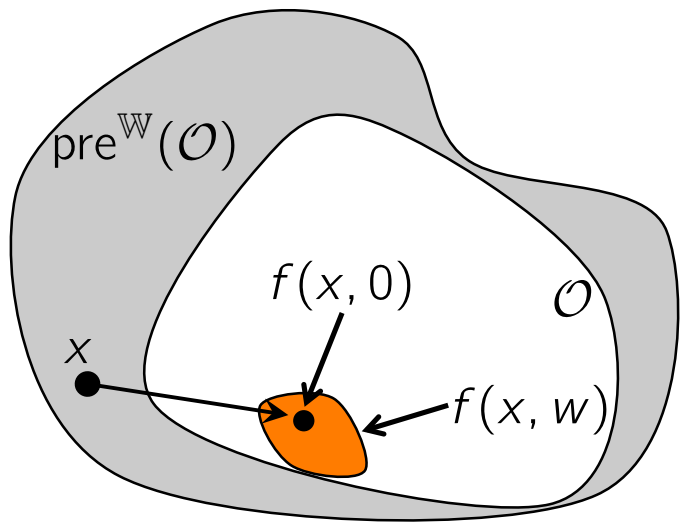
where $h_{\mathbb{W}}$ is the *support function* and $\ominus$ is called the Pontryagin difference.

Robust Invariant Set Conditions

Theorem: Geometric condition for robust invariance

A set $\mathcal{O}$ is a robust positive invariant set if and only if

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



Computing Robust Invariant Sets

Conceptual Algorithm to Compute Robust Invariant Set

Input: f, $\mathbb{X}$, $\mathbb{W}$
Output: $\mathcal{O}_\infty^{\mathbb{W}}$

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

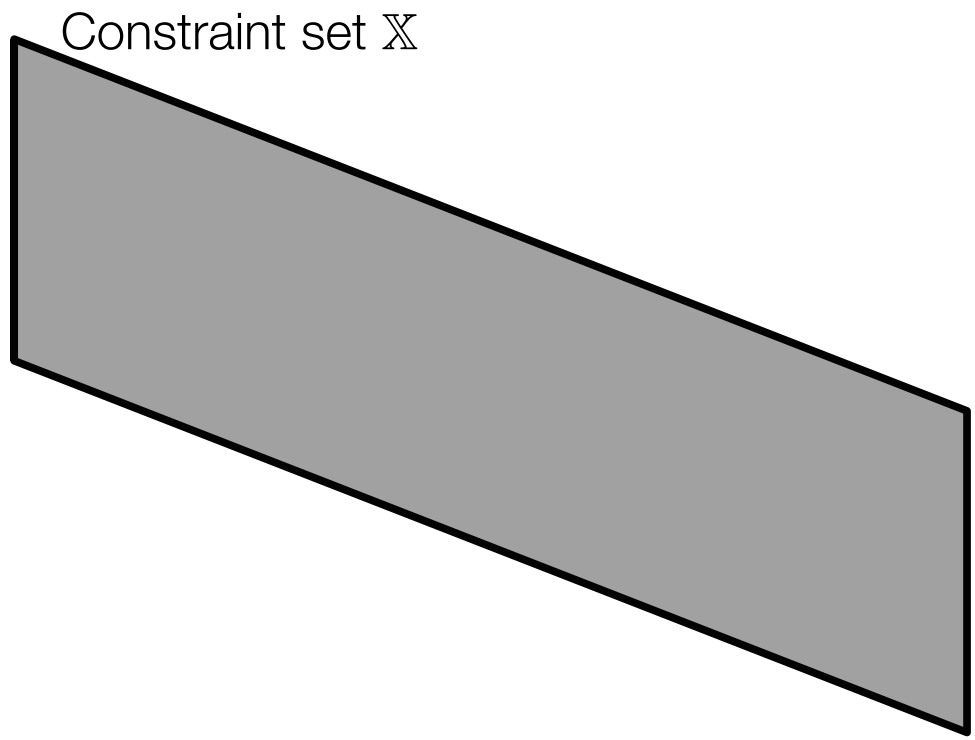
This is the same as for the nominal case, with $\text{pre}(\Omega)$ replaced by $\text{pre}^{\mathbb{W}}(\Omega)$.

Computing Robust Invariant Sets

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

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

K is the LQR controller for $Q = 0.1I$, $R = 1$

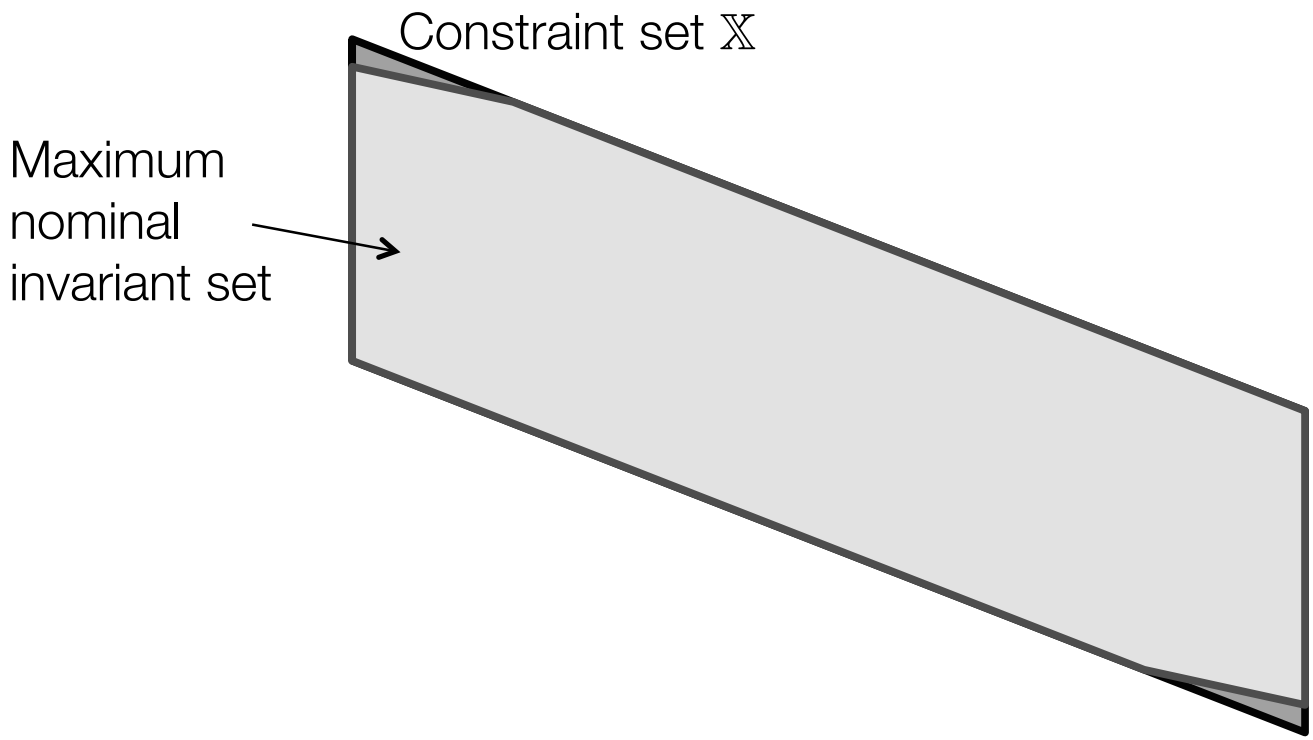


Computing Robust Invariant Sets

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

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

K is the LQR controller for $Q = 0.1I$, $R = 1$

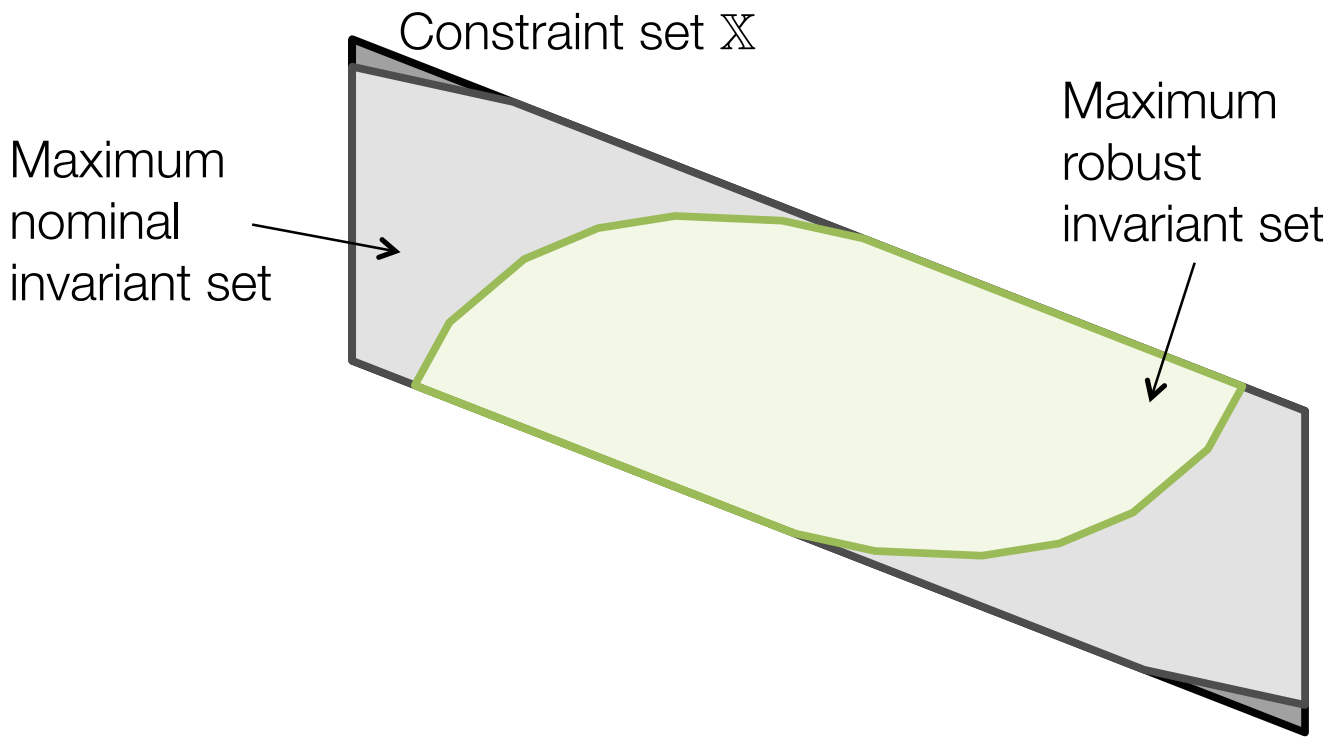


Computing Robust Invariant Sets

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

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

K is the LQR controller for $Q = 0.1I$, $R = 1$

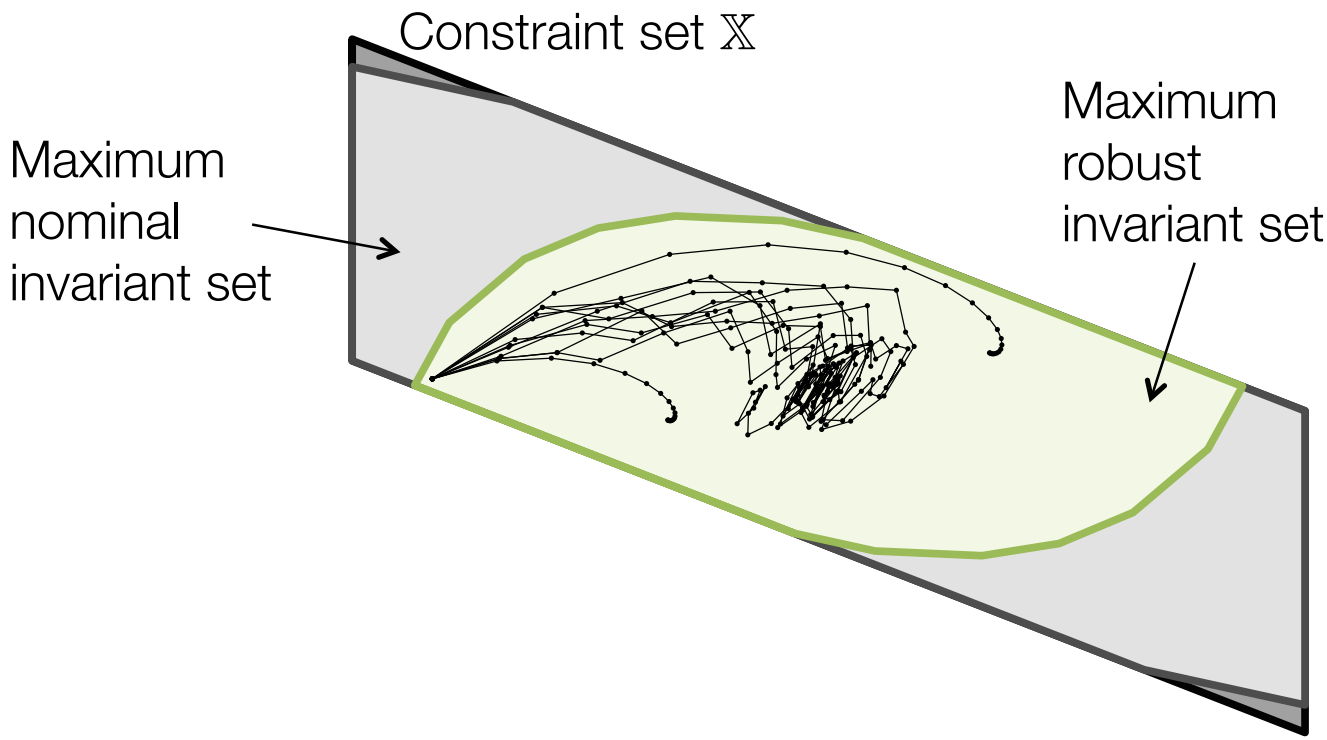


Computing Robust Invariant Sets

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

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

K is the LQR controller for $Q = 0.1I$, $R = 1$

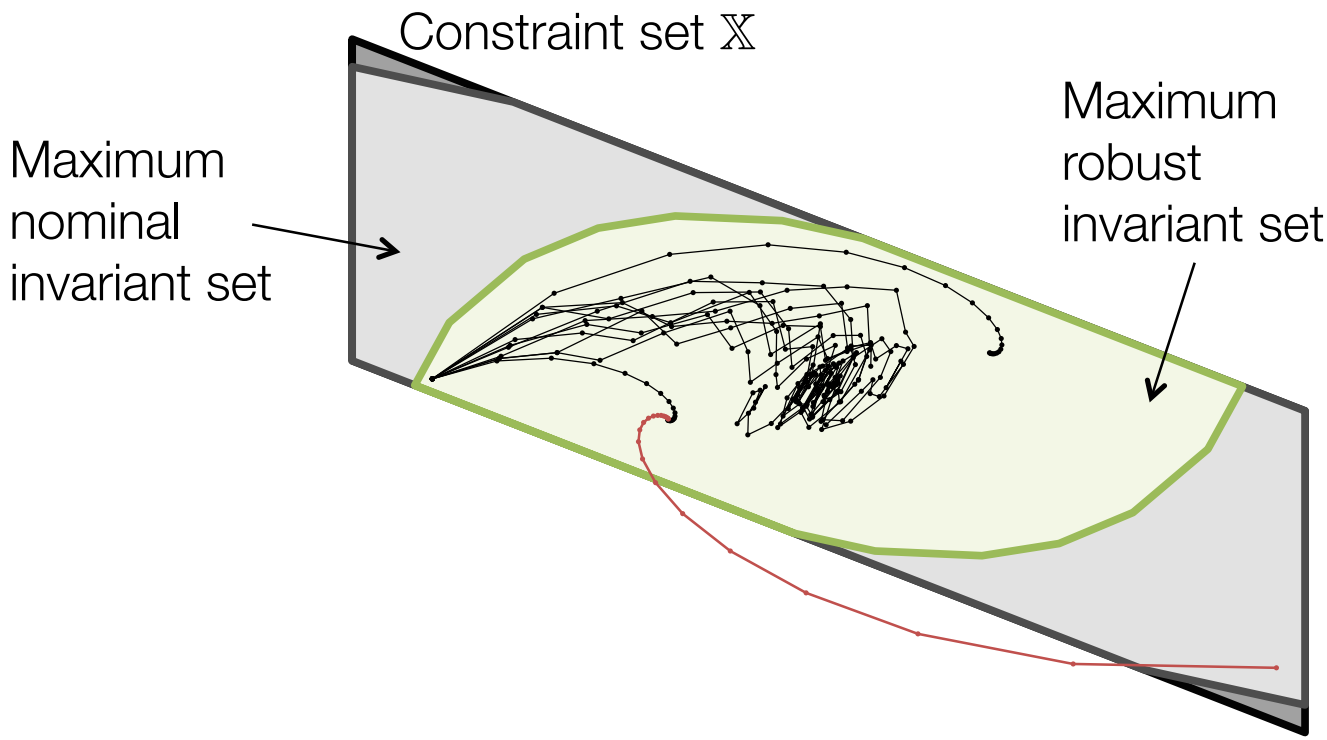


Computing Robust Invariant Sets

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

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

K is the LQR controller for $Q = 0.1I$, $R = 1$



Robust Constraint Satisfaction

Recall: We break the MPC prediction into two parts.

$$\left.\begin{aligned} \phi_{i+1} &= A\phi_i + Bu_i + w_i \\ u_i &\in \mathbb{U} \\ \phi_i &\in \mathbb{X}\ \forall \mathbf{w} \in \mathbb{W}^N \end{aligned}\right\}$$

- $i = 0, \ldots, N-1$
- Optimize over control actions $\{u_0, \ldots, u_{N-1}\}$
- Enforce constraints explicitly by imposing $\phi_i \in \mathbb{X}$ and $u_i \in \mathbb{U}$ for all sequences $\mathbf{w}$

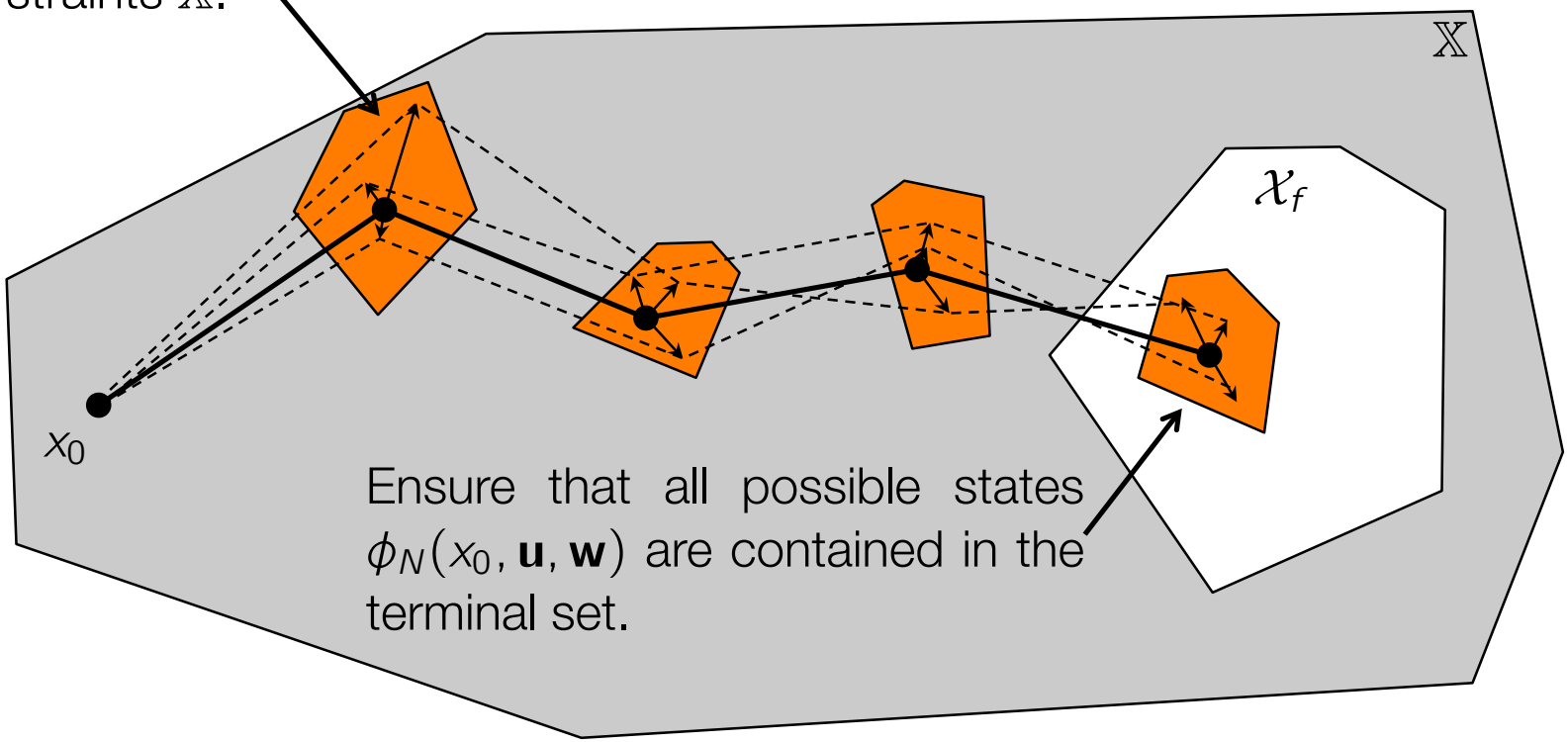
$$\left.\begin{aligned} \phi_{i+1} &= (A + BK)\phi_i + w_i \\ \phi_N &\in \mathcal{X}_f \end{aligned}\right\}$$

- $i = N, \ldots$
- Assume control law to be linear $u_i = K\phi_i$
- Enforce constraints implicitly by constraining ϕ_N to be in an *robust invariant set* $\mathcal{X}_f \subseteq \mathbb{X}$ and $K\mathcal{X}_f \subseteq \mathbb{U}$ for the system $\phi^+ = (A + BK)\phi + w$

In the following:

- Robustly enforcing constraints of a linear system
- **Robustly ensuring constraints of the sequence** $\phi_1, \ldots, \phi_{N-1}$

Robust Constraint Satisfaction



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

Uncertain State Evolution

Nominal system

$$x^+ = Ax + Bu$$

$$x_1 = Ax_0 + Bu_0$$
$$x_2 = A^2x_0 + ABu_0 + Bu_1$$
$$\vdots$$
$$x_i = A^ix_0 + \sum_{k=0}^{i-1} ABu_{i-k}$$

Uncertain system

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

$$\phi_1 = Ax_0 + Bu_0 + w_0$$
$$\phi_2 = A^2x_0 + ABu_0 + Bu_1 + Aw_0 + w_1$$
$$\vdots$$
$$\phi_i = A^ix_0 + \sum_{k=0}^{i-1} A^kBu_{i-k} + \sum_{k=0}^{i-1} A^kw_{i-k}$$
$$\phi_i = x_i + \sum_{k=0}^{i-1} A^kw_{i-k}$$

Goal: Ensure $\phi_i \in \mathbb{X}$ for all $\mathbf{w} \in \mathbb{W}^N$

Robust Constraint Satisfaction

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

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

Assume that $\mathbb{X} = \{x \mid Fx \leq f\}$, then this is equivalent to

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

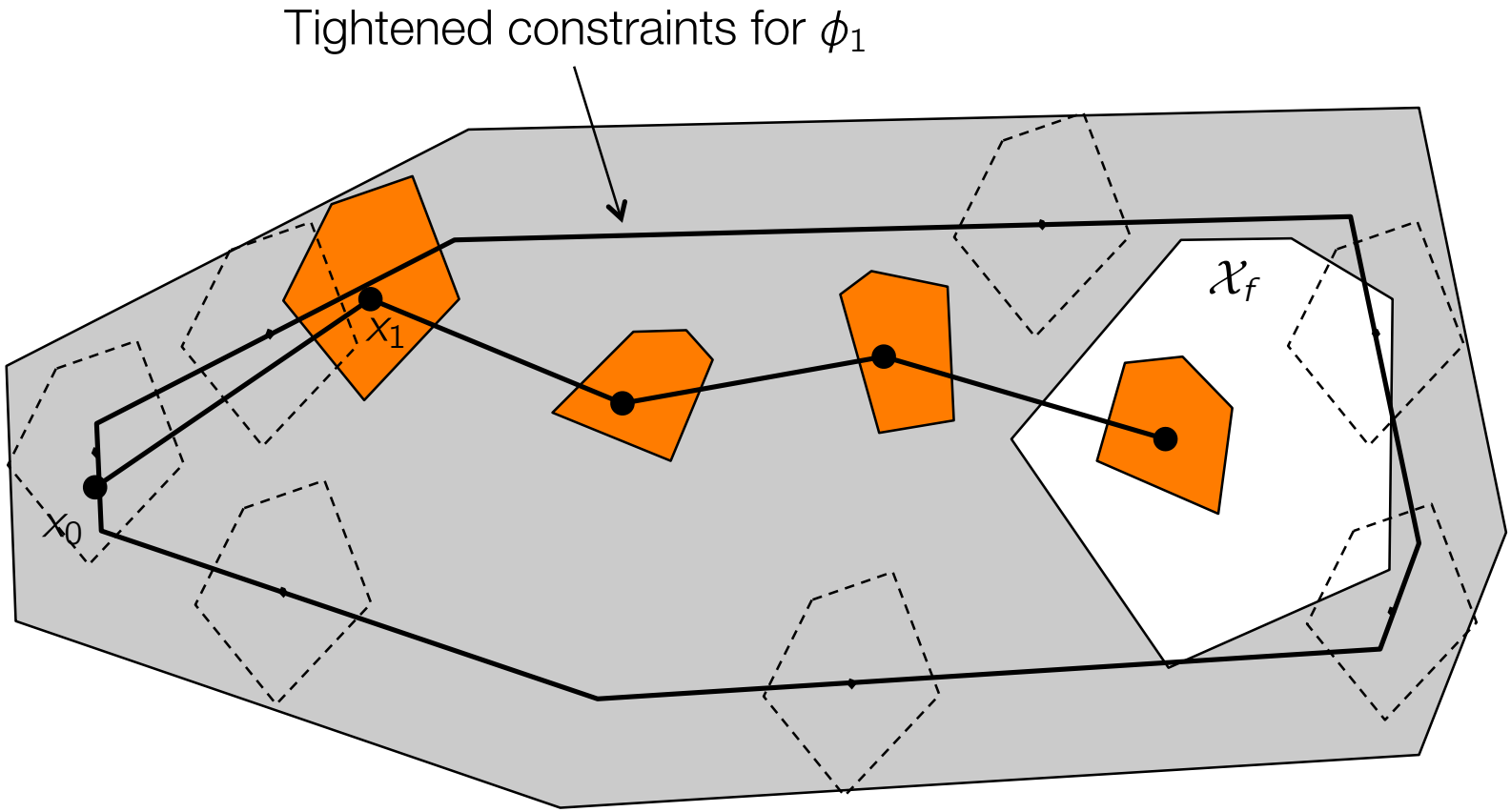
We've seen this before while computing the robust pre-set:

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

The support function can be pre-computed offline.
All we're doing is tightening the constraints on the nominal system

Robust Constraint Satisfaction

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



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

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

Terminal State Constraint

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 $\mathcal{X}_f$:

$$\phi_N(x_0, \mathbf{u}, \mathbf{w}) \subseteq \mathcal{X}_f$$

This is handled in exactly the same fashion.

Outline

1. Uncertainty Models

2. Impact of Bounded Additive Noise
   - Choosing a cost to minimize
   - Robust Constraint Satisfaction

3. Robust Open-Loop MPC

Goals of Robust Constrained Control

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

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

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

Putting it Together

Robust Open-Loop MPC

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

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

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

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

Properties of Robust Open-Loop MPC

Robust Control Invariance
If $\mathbf{u}^\star(x)$ is the optimizer of the robust open-loop MPC problem, then the system $Ax + Bu_0^\star(x) + w \in \mathbb{X}$ 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.

We have shown the key property of robust MPC: robust invariance.

However, we have not shown convergence...

Goals of Robust Constrained Control

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

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

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

Analysis is not direct - we will consider this in a more general setting next week.

Goals of Robust Constrained Control

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

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

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

Robust open-loop MPC has a *very* small region of attraction!
This is why **you should not use it!**
It is a theoretical development to give you the concepts. Next week we will go through some more practical methods.

Lecture Take Homes

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

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

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

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