MECH 6v29.002 – Model Predictive Control

Tuesday and Thursday 8:30 – 9:45am

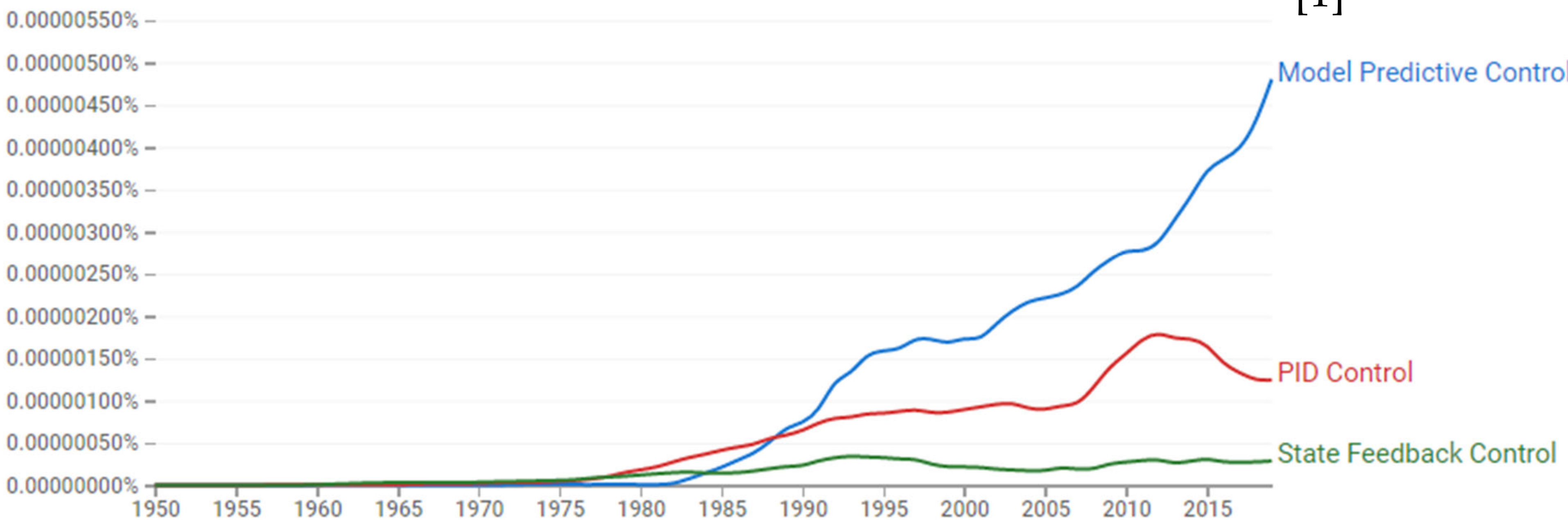
L2 – Key Concepts

Outline

- History
- MPC vs PID
- Key concepts
- Motivating example
- Who am I? – History/experience with MPC

Brief History

[1]

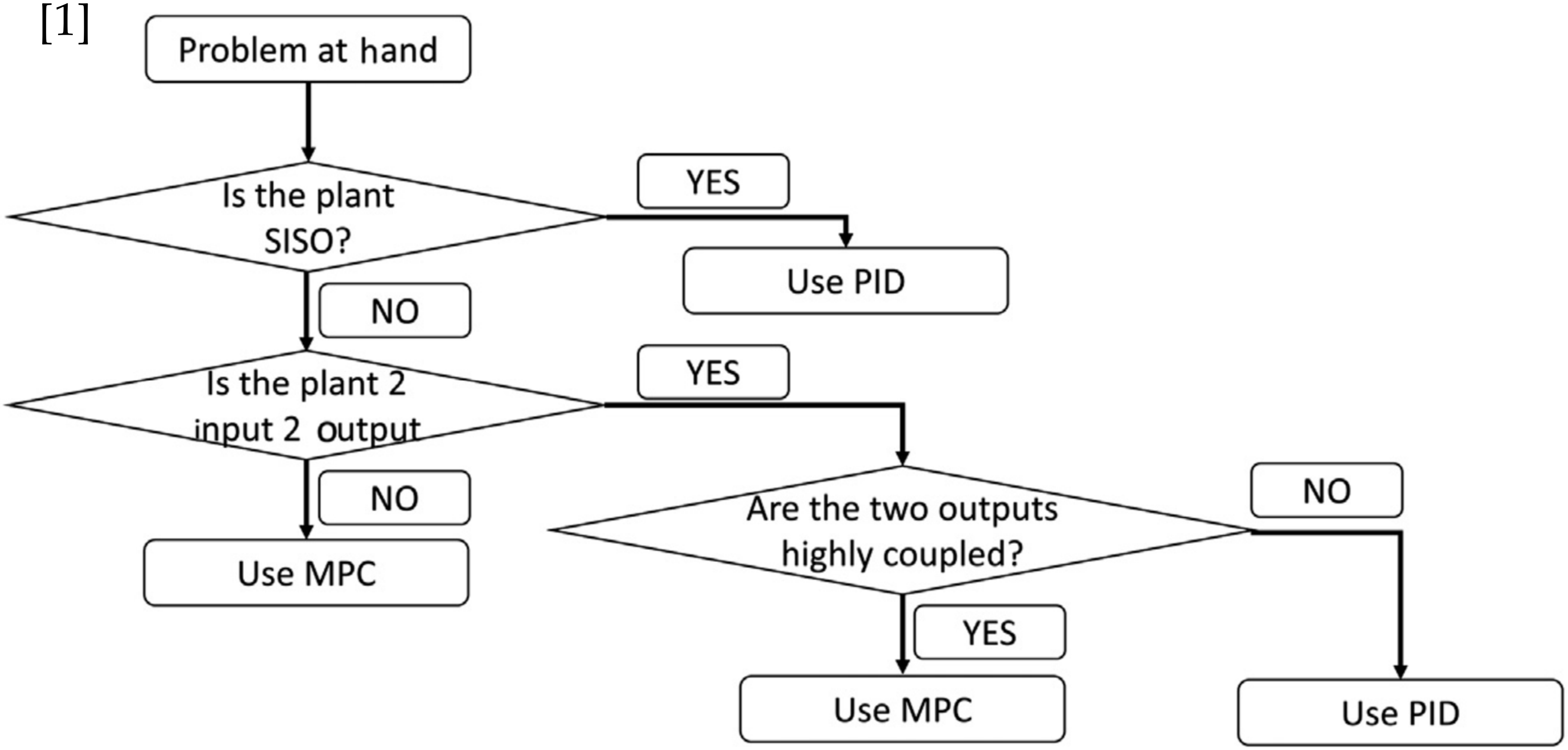


- Significant increase in popularity in last 40 years
- Also known as Dynamical Matrix Control (DMC), Generalized Predictive Control (GPC), or Receding Horizon Control (RHC)
- First applications come from Shell Oil in 1980 for refineries.
- Now there are too many applications to count
  - Largely due to advances in computational power

[1] Practical Design and Application of Model Predictive Control by Nassim Khaled and Bibin Pattel, 2018

MPC vs PID

[1]



- PID is still the first choice in many industries
  - Easy to understand and tune
  - Many ad hoc approaches to overcome limitations

[1] Practical Design and Application of Model Predictive Control
by Nassim Khaled and Bibin Pattel, 2018

Intro to Model Predictive Control

- Mathematical Formulation

Prediction horizon length: N
Input sequence: $U_t = \{u_t, u_{t+1}, \ldots, u_{t+N-1}\}$

Cost Function to be minimized via optimization

$$\min_{U_t} \sum_{k=t}^{t+N-1} J(x_k, u_k) + J_T(x_{t+N}, u_{t+N})$$

subject to:

$$s.t.\ \forall k = t, \ldots, t+N-1$$

- Dynamic system constraints (typically state-space model)

$$x_{k+1} = f(x_k, u_k),$$

- Input, state, and output constraints

$$u_k \in \mathcal{U},\ x_k \in \mathcal{X},\ y_k = g(x_k, u_k) \in \mathcal{Y}$$

- Initial condition constraint

$$x_t = x(t)$$

1. Measure current state: $x(t)$
2. Solve for optimal input sequence: U_t^*
3. Only apply first input: u_t^*
4. Repeat at t+1

When to use MPC

In no particular order:

- When you are not a control expert – it is pretty intuitive / easy to tune
- Applicable to a wide variety of systems ranging from simple to complex, systems with delays, nonminimum phase, unstable
- MIMO – approach is unaffected by number of inputs/states/outputs
- Can directly compensate for time delays
- Can provide feedforward-type action when future disturbances or references can be estimated
- "Easy to implement" – depends on what is "easy"
- Constraints are systematically integrated in the design process
- MPC is an idea/approach based on a certain set of basic principles
    - There is a lot of room for customization to fit application

Intro to Model Predictive Control

- Key issues

  Planned open-loop trajectories
  $\neq$
  closed-loop trajectories

  - even if you have a perfect model and no disturbances
  - often due to finite prediction horizon
- Feasibility
  - How can we guarantee that the optimization problem will always have at least one solution that satisfies all constraints?
- Stability
  - How can we guarantee the closed-loop system is stable?
  - Not as easy as checking eigenvalues since we are solving an optimization problem at each time-step.
- Constraint satisfaction
  - How can we guarantee that the closed-loop state and input trajectories satisfy the desired constraints?
- Real-time implementation
  - Will we be able to solve the optimization problem fast enough?

Drawbacks

- Most of the time*, it requires solving an optimization problem "online" = in real time

  * the unconstrained case and explicit MPC are notable exceptions

- Even though computers are getting better, there are still plenty of applications where the computational resources available are extremely limited
  - Can you think of any?
- Need a "good" model of the system in an "appropriate" form

Key Concepts of MPC[1]

- Prediction
- Receding horizon
- Modeling
- Performance index (cost function)
- Constraint handling
- Multivariable

We will summarize each of these today and then discuss most of these in more detail throughout the semester

[1] See John Rossiter's Youtube page for 400+ videos on dynamic system modeling and control

Prediction

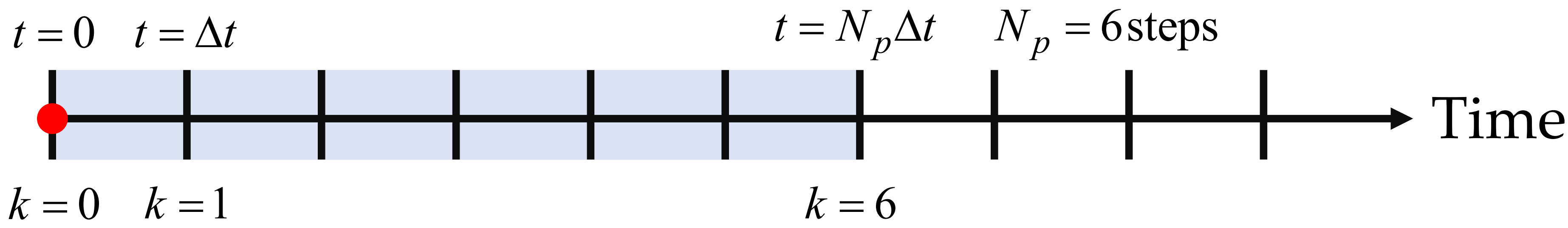
- What can you do with prediction?
- How far should we predict?
  - Is there such a thing as too short or too long of a prediction horizon?
- How do we achieve this prediction?
- How important is it to have perfect predictions?

General ideas:

- Need a prediction horizon that goes beyond the settling time of a dynamic system (captures the transient behavior)
- Too long can add computational cost without performance improvement
- The required accuracy of the model depends on what you are trying to do, but you can sometimes get very effective control with a rather poor model (due to feedback)

Receding Horizon

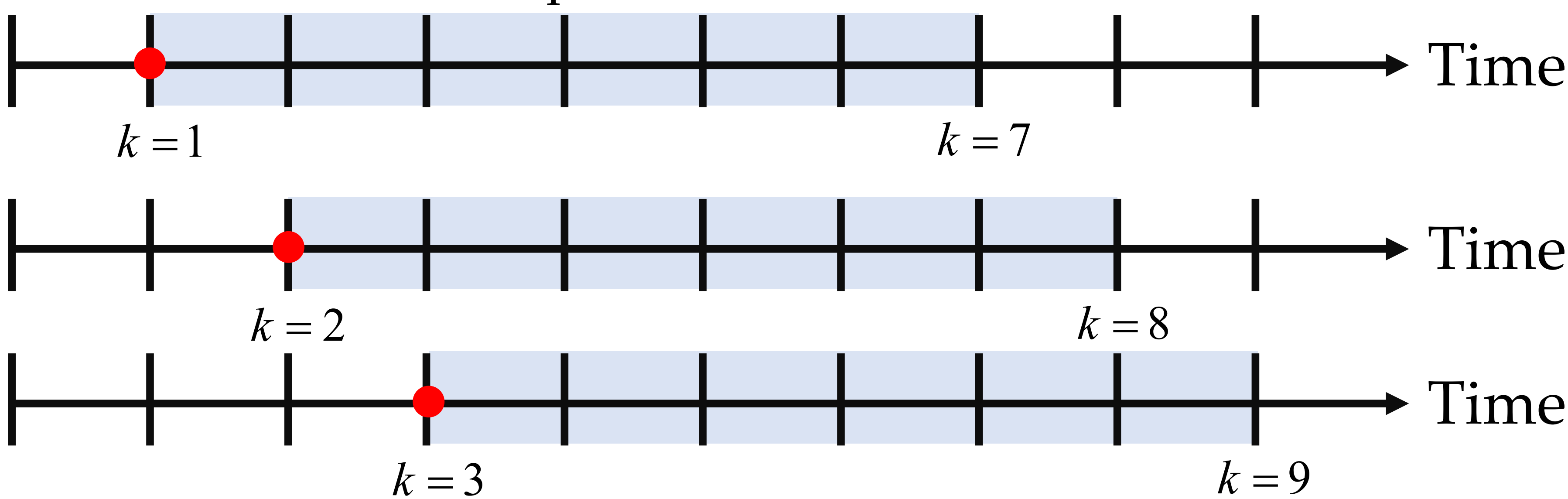
Prediction into the future



- Take a measurement $x(0)$
- Apply optimal input $u^*(0)$
- Move horizon forward and repeat

Input is held piecewise-constant

$$u(t)=u^*(0),\ \forall t\in[0,\Delta t)$$



Δt Discrete time step size

● Current time step

Finite prediction horizon

The repeated measurement and re-planning introduces feedback!

Modeling

- Need a mathematical model that captures the (dynamic) relationship between future inputs/state/outputs of the system – this enables prediction
- Model considerations
  - "All models are wrong, but some are useful." [1]
  - "Everything should be made as simple as possible, but no simpler." – Einstein
  - Modeling can be over 50% of the control design effort
- Ideally a model is
  - Easy to use to form predictions
  - Results in a optimization problem that is easy to solve
  - Gives "accurate predictions"
    - What does accurate mean? Steady-state, transients, modes?
    - 10-20% model error is typically easy to overcome via feedback

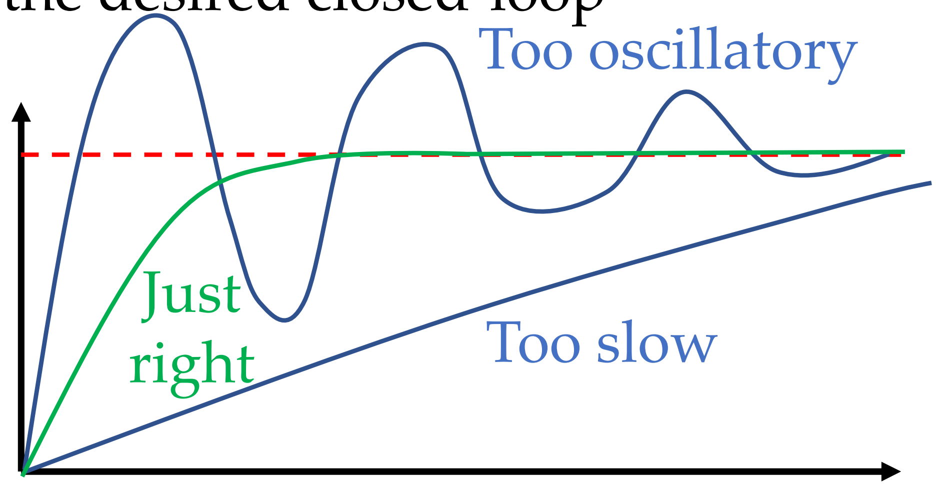
Linear model

Nonlinear model?

[1] https://en.wikipedia.org/wiki/All_models_are_wrong

Performance Index (Cost Function)

- The cost function provides a quantitative way to capture the "cost" of operating the system.
- Can be designed and tuned to create the desired closed-loop behavior
- This is often a challenge
  - You typically know what good control performance looks like
  - But this can be hard to quantify in an appropriate way
  - Other applications might have higher-level control objectives
    - Balancing robot – don't fall over
    - But how do you design a cost function to quantify this?
  - There might be many conflicting/competing objective
    - Performance, efficiency, safety.



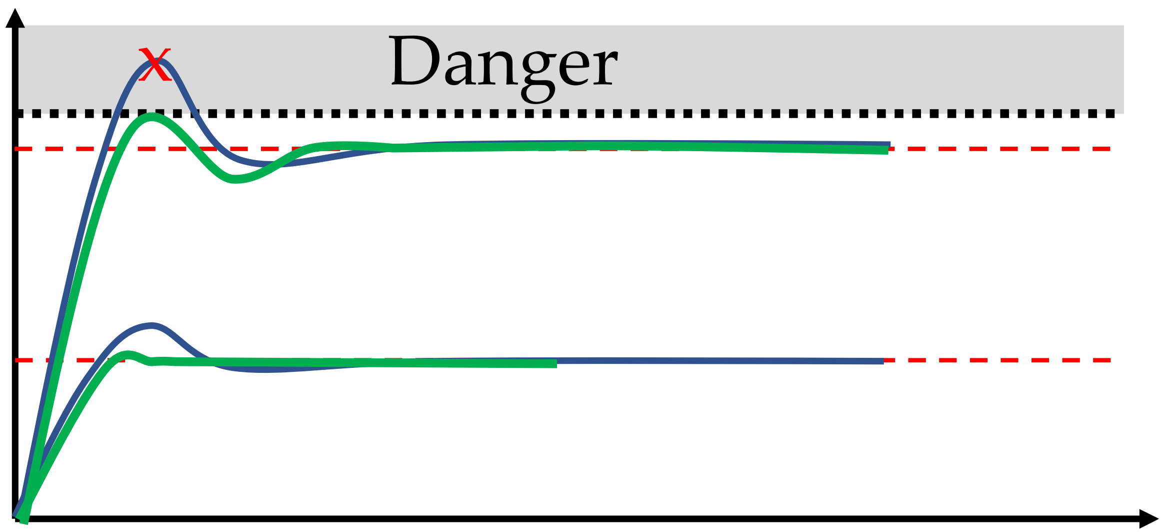
Perhaps looking for a desired damping ratio

But cost function is of the form:

$$J(k) = x_k^T Q x_k + u_k^T R u_k$$

Constraint Handling

- MPC allows you to embed constraints directly in the optimization problem
- Therefore, control inputs are "optimal" with respect to the imposed constraints
- While MPC provides a systematic way of directly accounting for constraints, most other control formulations need to be modified in some ad hoc way to handle constraints as an after thought
- Example:
    - A well-tuned PID controller
    - MPC knows about these constraints and can adjust control accordingly



Multivariable

- MPC is capable of handling multiple inputs/states/outputs
- No difference in MPC control design
- Each input will likely affect multiple state/outputs
    - MPC knows of all these interactions and optimizes with respect to all of them simultaneously
    - Similar to LQR in this way
    - PID (multiple SISO loops) generally ignores this coupling and requires “weak” coupling between loops

Motivating Example

- With a little practice and a general understanding of these key concepts, MPC will often work in practice
  - Can be relatively easy to get a model and a working, well-tuned controller in simulation
  - However, there are many reasons why it might not work, and the theory associated with MPC can help a lot!
- Let's see some of these challenges first hand through and example and we will later study the theory used to overcome these challenges
- Linear system (transfer function):

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25}, \qquad \omega_n = 0.5 \quad \zeta = 0.2 \quad G(0) = 1$$

$$p = -0.1 \pm 0.5i \qquad z = \frac{1}{15} = 0.0667$$

- State-space model (Observable Canonical Form):

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

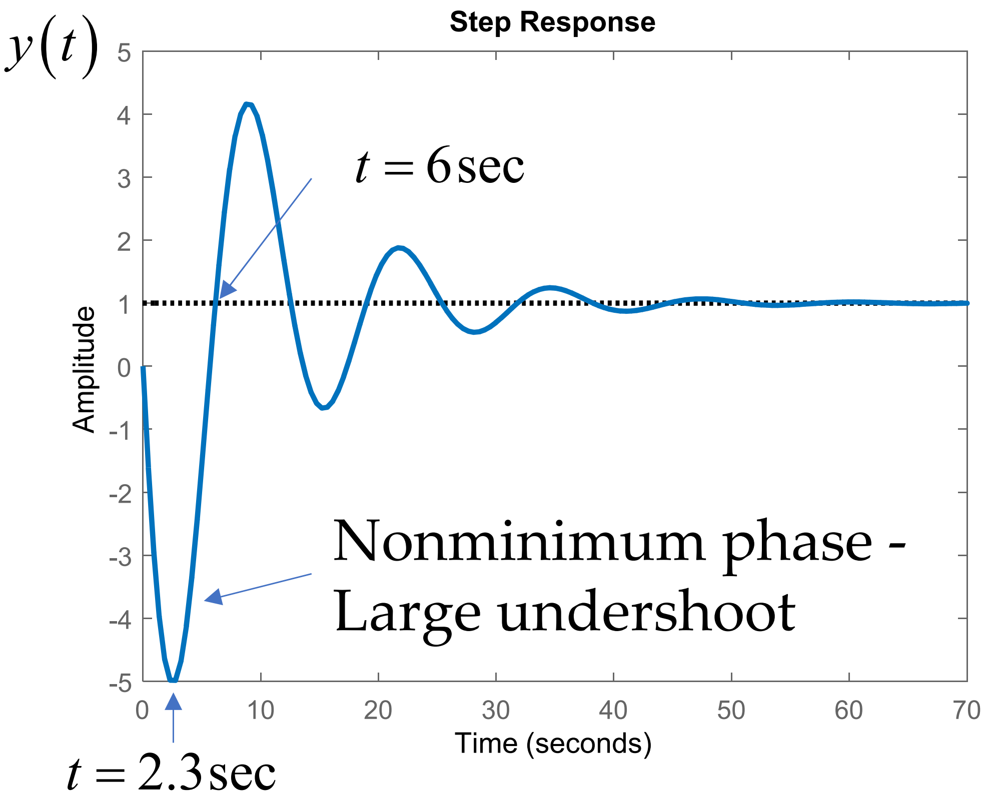
$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

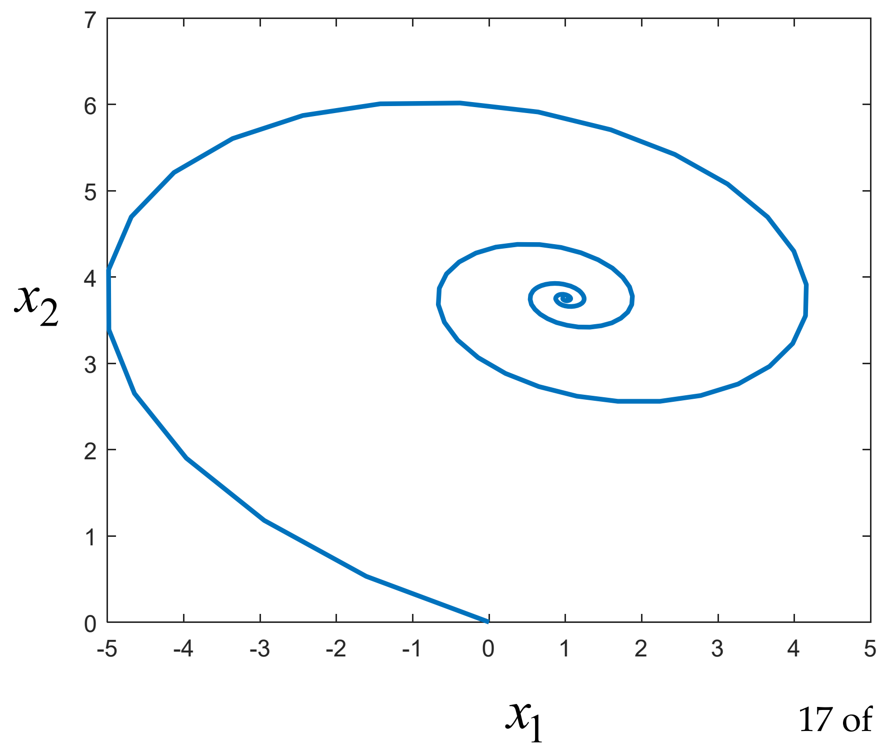
$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

- Unit step response:





Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

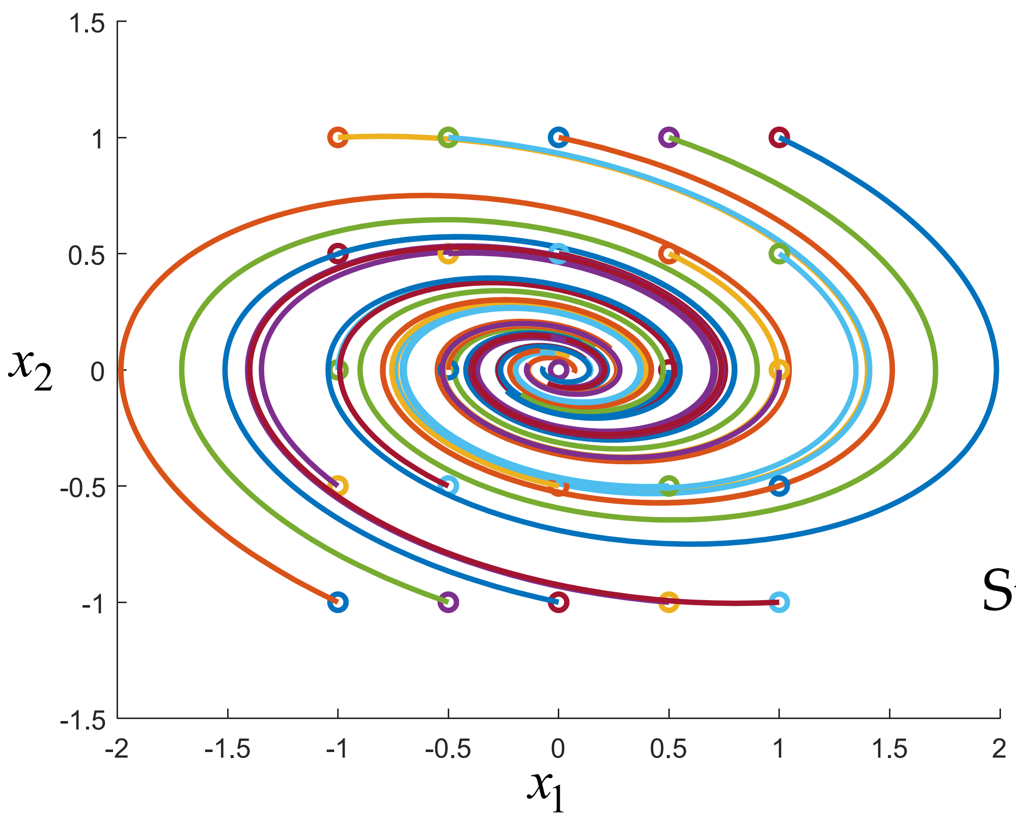
$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

- Free response from non-zero initial conditions



Stable, oscillatory system

Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

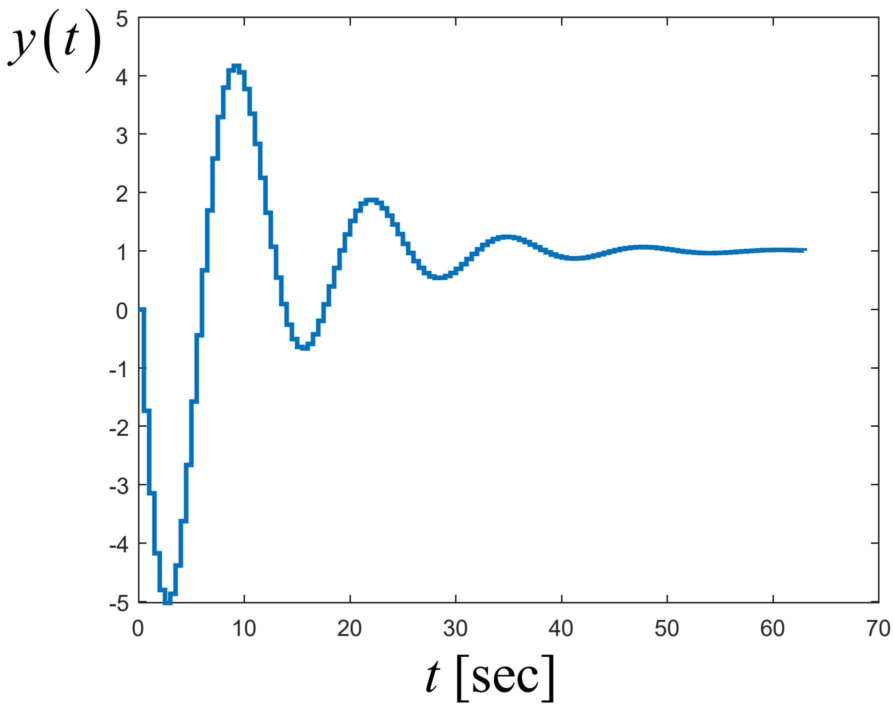
$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

$$A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

- MPC control design
  - Discretized model with $\Delta t = 0.5\,\text{sec}$

$$x_{k+1} = A_d x_k + B_d u_k$$
$$y_k = C_d x_k + D_d u_k$$



Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = [0]$$

- MPC control design

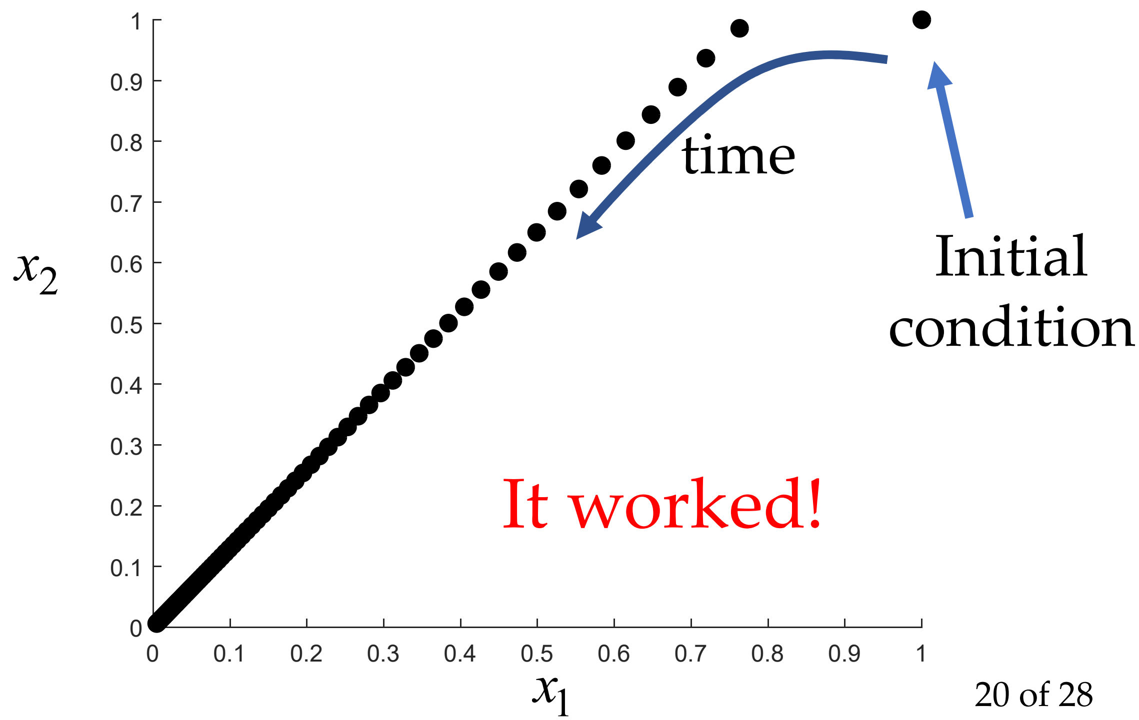
$$N_p = 10 \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad R = [0.1]$$

$$\min_{U_k} \sum_{k=0}^{N_p} x_k^{\,T} Q x_k + u_k^{\,T} R u_k$$

Subject to:

$$x_{k+1} = A_d x_k + B_d u_k$$
$$x_0 = x(t)$$

Goal: Drive the system to the origin (0,0)



Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

- MPC control design

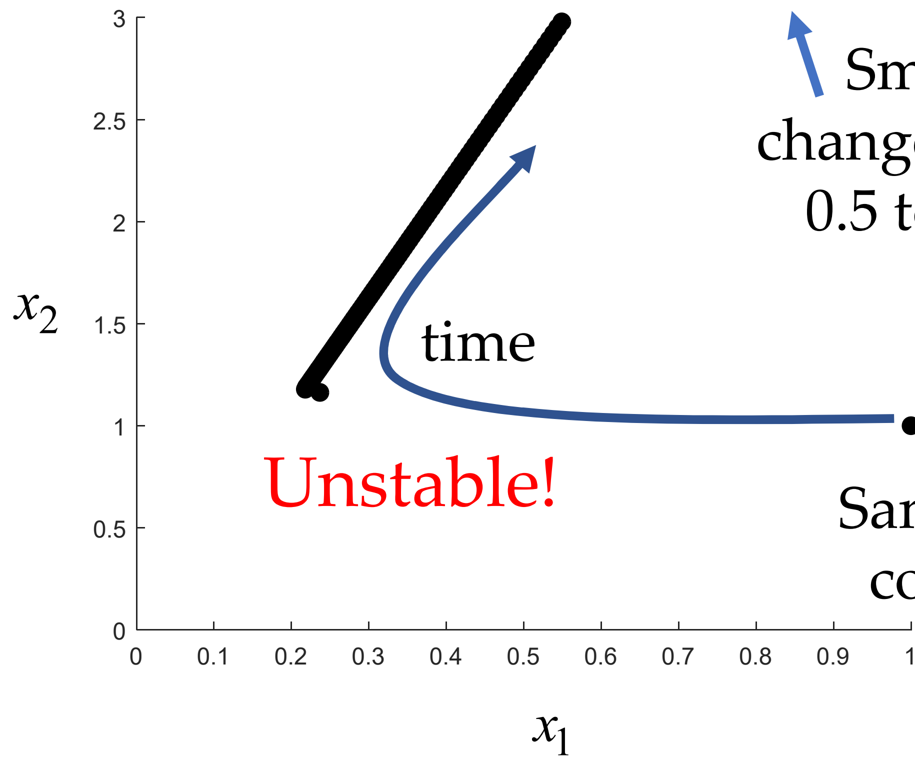
$$\min_{U_k} \sum_{k=0}^{N_p} x_k^T Q x_k + u_k^T R u_k$$

Subject to:

$$x_{k+1} = A_d x_k + B_d u_k$$
$$x_0 = x(t)$$

$$N_p = 10 \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad R = [0.1]$$

Small change from 0.5 to 0.1



Same Initial condition

Goal: Drive the system to the origin (0,0)

Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$

$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = [0]$$

- MPC control design

$$\min_{U_k} \sum_{k=0}^{N_p} x_k{}^T Q x_k + u_k{}^T R u_k$$

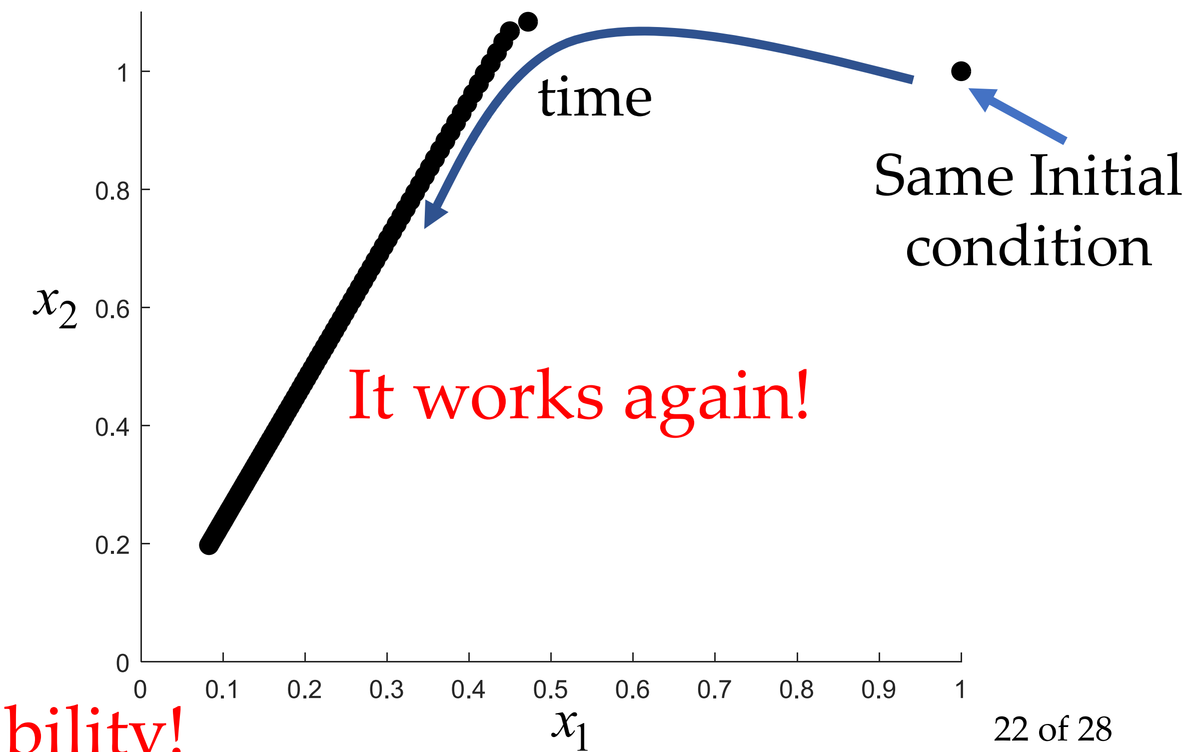
Subject to:

$$x_{k+1} = A_d x_k + B_d u_k$$
$$x_0 = x(t)$$

Increase horizon from 10 to 20

$N_p = 20 \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad R = [0.1]$

Goal: Drive the system to the origin (0,0)



Need theory to guarantee stability!

Motivating Example

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.25(-15s+1)}{s^2+0.2s+0.25},$$

$$\omega_n = 0.5 \qquad \zeta = 0.2 \qquad G(0) = 1$$

$$p = -0.1 \pm 0.5i \quad z = \frac{1}{15} = 0.0667$$

$$\dot{x} = Ax + Bu \qquad A = \begin{bmatrix} 0 & 1 \\ -0.25 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} -3.75 \\ 1 \end{bmatrix}$$

$$y = Cx + Du \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

- MPC control design

$$\min_{U_k} \sum_{k=0}^{N_p} {x_k}^T Q x_k + {u_k}^T R u_k$$

Subject to:

$$x_{k+1} = A_d x_k + B_d u_k$$

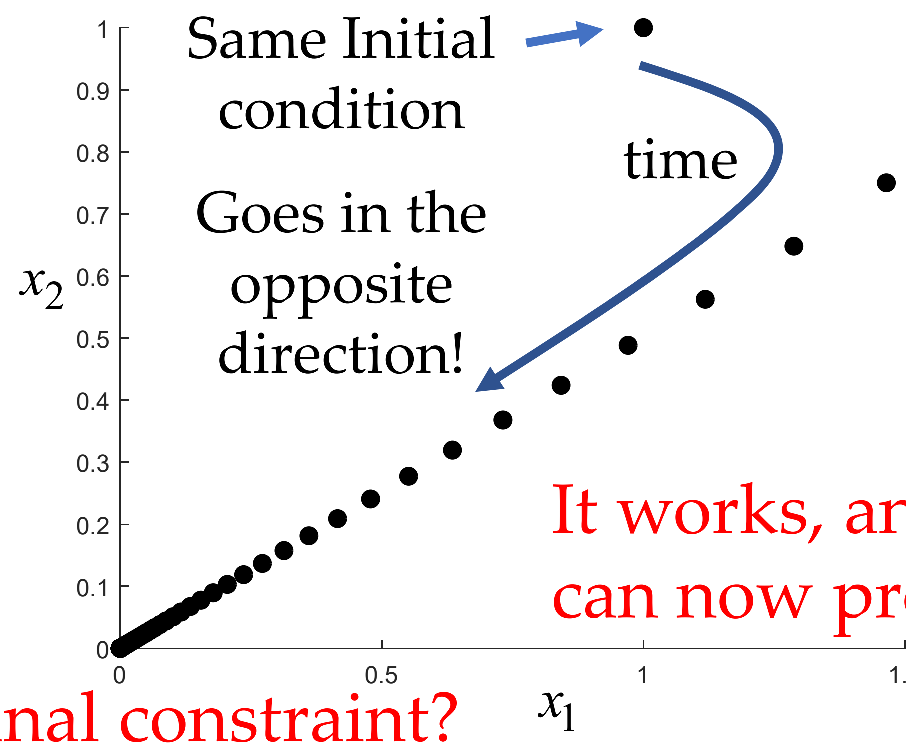
$$x_0 = x(t)$$

$$x_{N_p} = 0$$

Add terminal constraint to force to origin

Decrease horizon back to 10

$$N_p = 10 \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad R = [0.1]$$



It works, and we can now prove it!

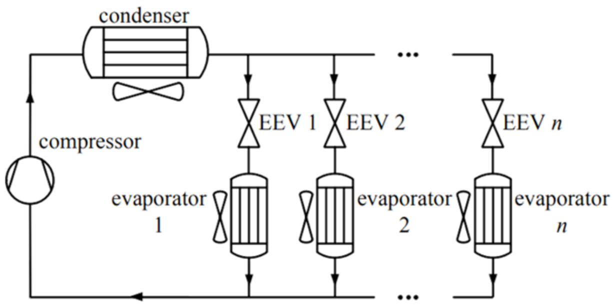
Any issues with adding terminal constraint?

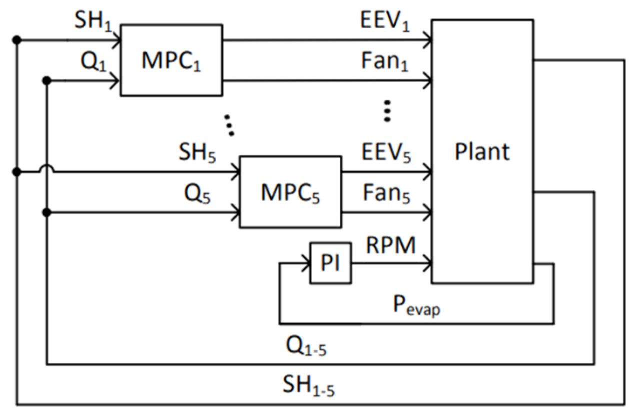
Who am I?

- History with MPC

2013: ACC - Decentralized MPC of multi-evaporator vapor compression systems

- Application driven – little analysis

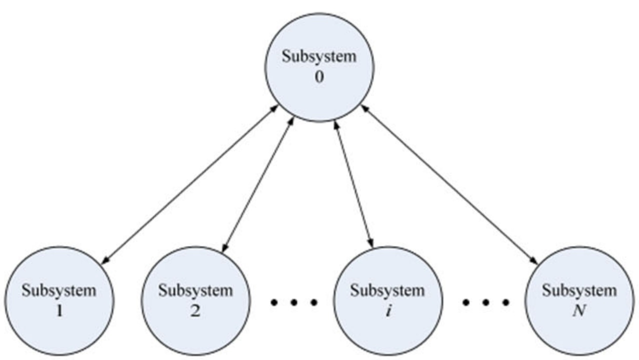




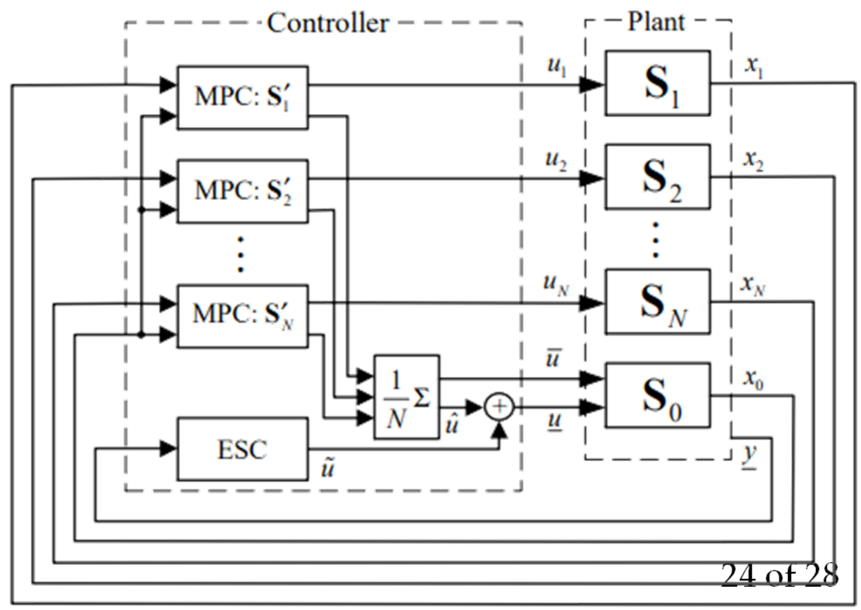
2013: MS Thesis - Decentralized MPC of multi-evaporator vapor compression systems

- Exploit Block Arrow Structure (BAS)

- Combine MPC and ESC



$$A = \left[\begin{array}{cccc|c} A_{11} & 0 & \cdots & 0 & A_{10} \\ 0 & A_{22} & \ddots & \vdots & A_{20} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & A_{NN} & A_{N0} \\ \hline A_{01} & A_{02} & \cdots & A_{0N} & A_{00} \end{array}\right]$$

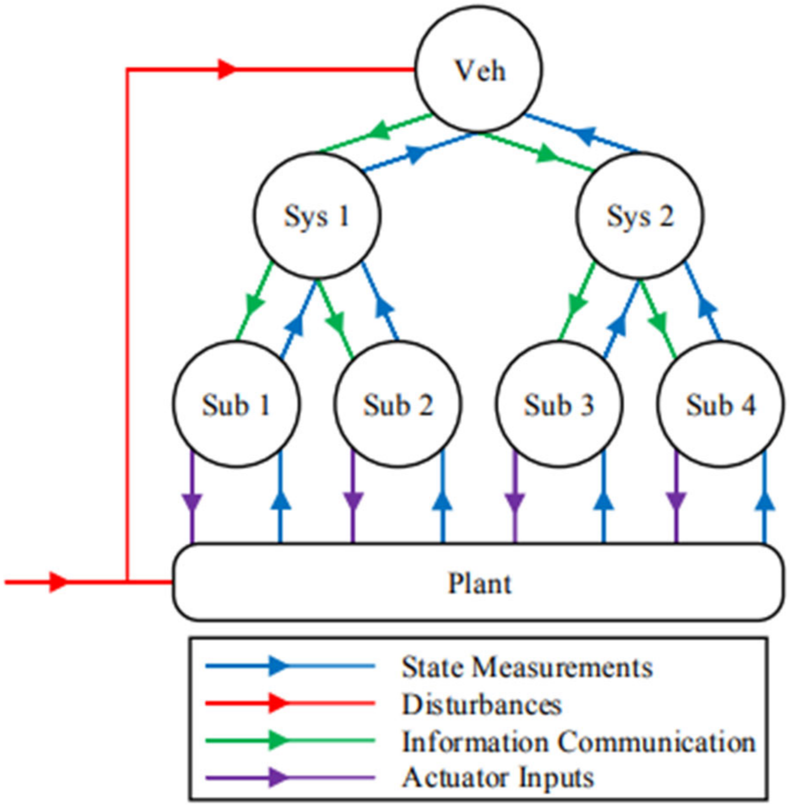


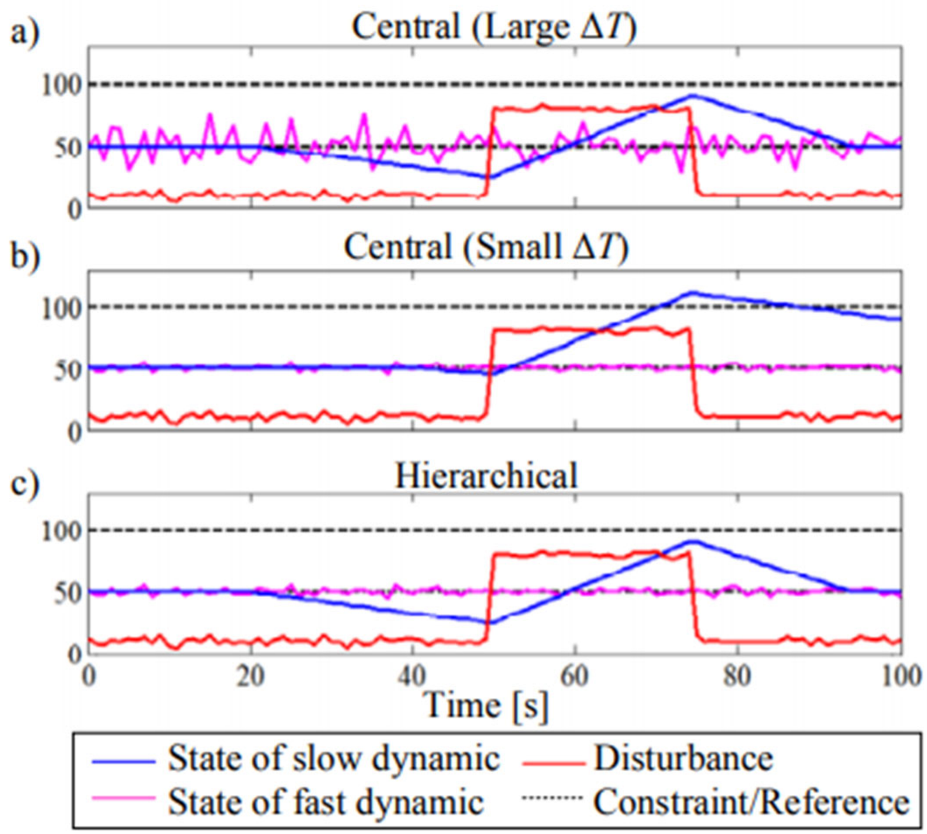
Who am I?

- History with MPC

2015: DSCC – Hierarchical MPC of systems using graph-based modeling

- Initial formulation – little analysis
- Motivation – match levels of hierarchy to each of the system's dynamic timescales





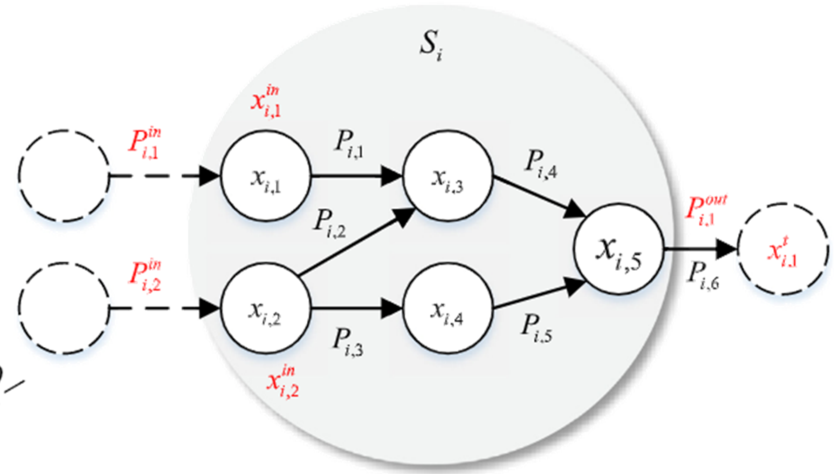
Who am I?

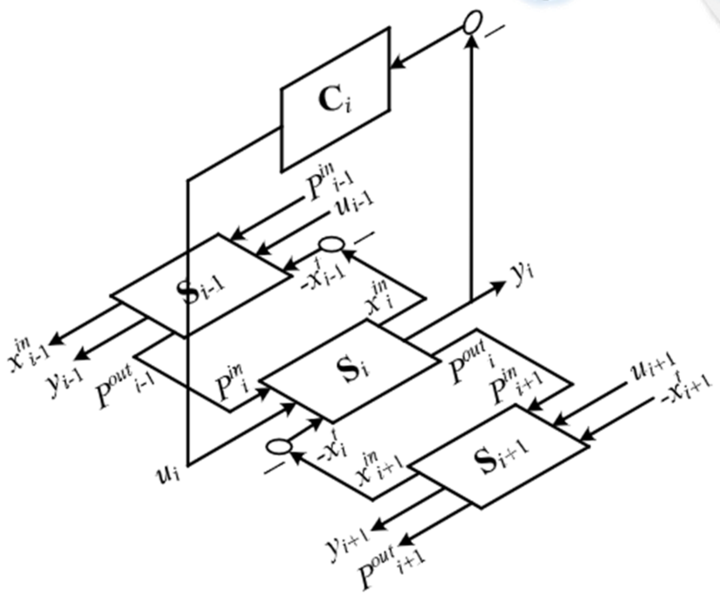
- History with MPC

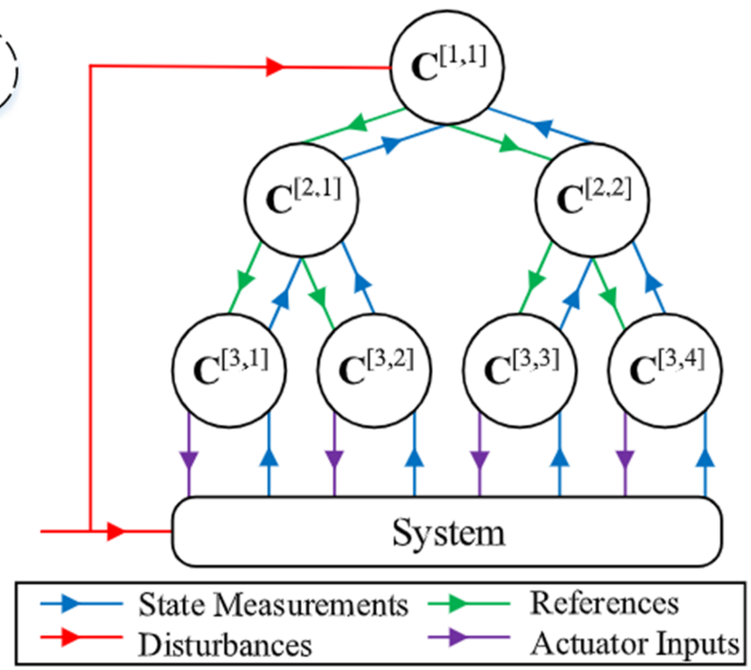
2017: Automatica – Stability of decentralized MPC of graph-based power flow systems via passivity

2018: Automatica – Robust hierarchical MPC of graph-based power flow systems

- Theory-driven – can we prove stability and constraint satisfaction







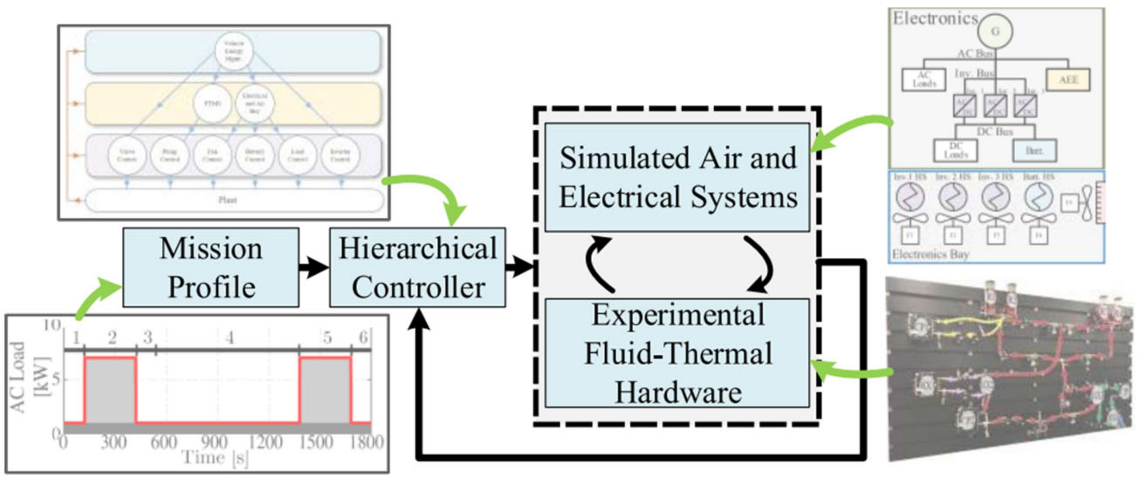
Who am I?

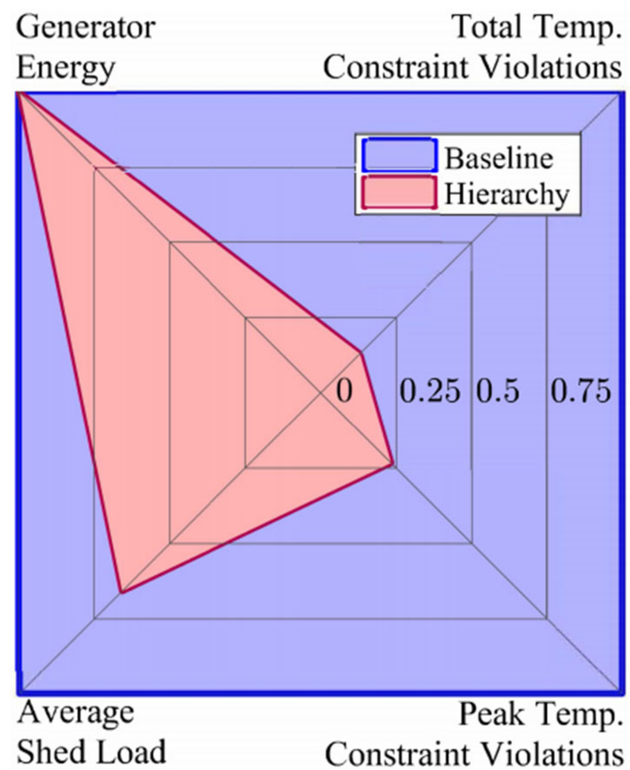
- History with MPC

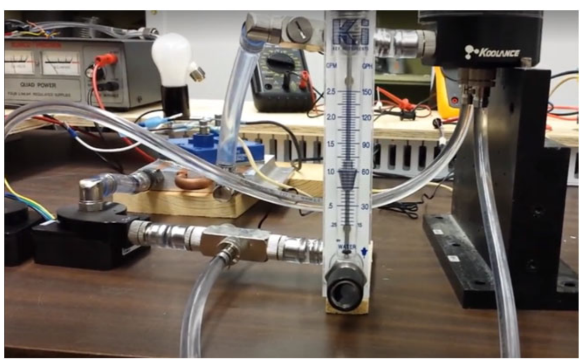
2019: TCST – Hierarchical control of aircraft electro-thermal systems

- Application to Hardware-In-the-Loop (HIL) system

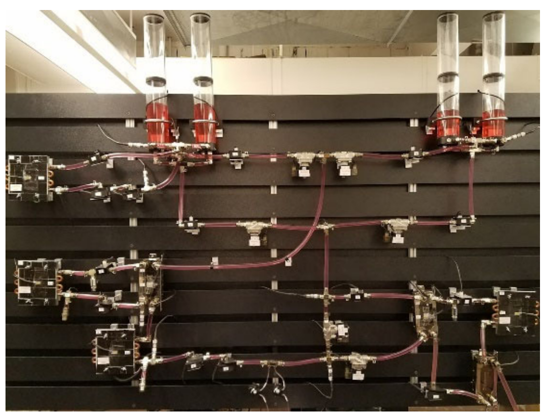
- Demonstration of hierarchical MPC performance for complex, multi-timescale, multi-energy domain system







July 29, 2014



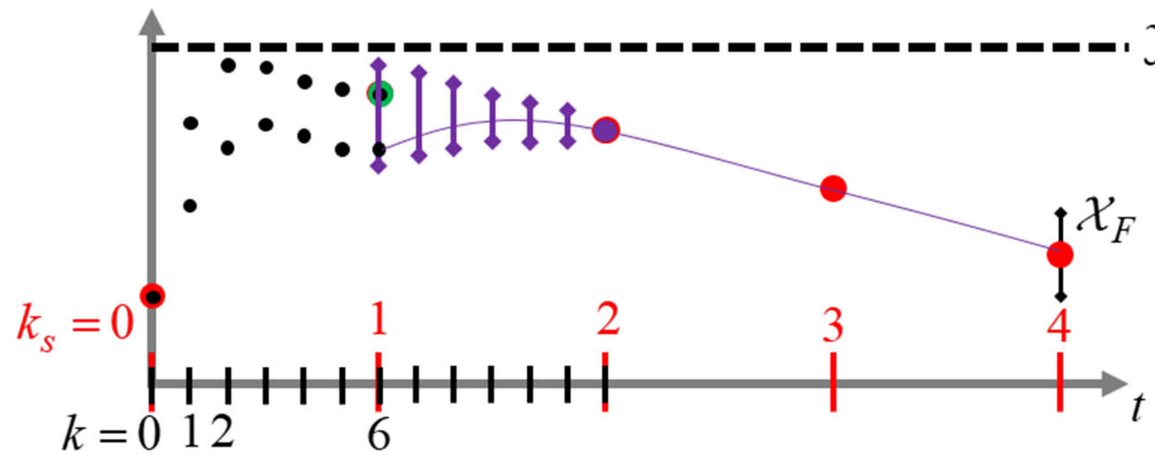
Oct. 8, 2017

Who am I?

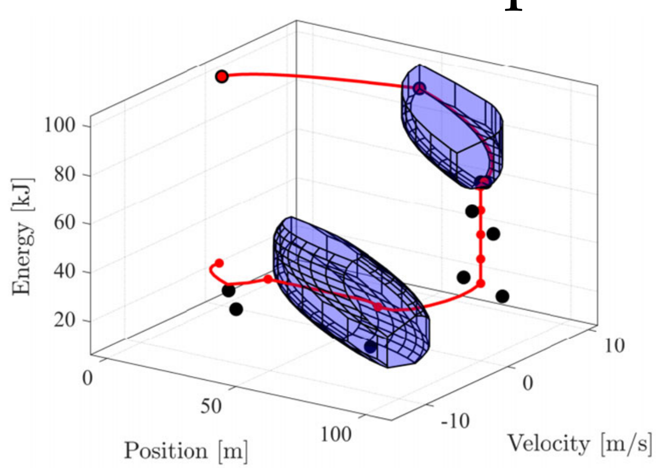
- History with MPC

Since joining UTD in Jan. 2018 with my Energy Systems Control Lab

Set-based Hierarchical MPC



Constrained zonotope waysets



Terminal Costs