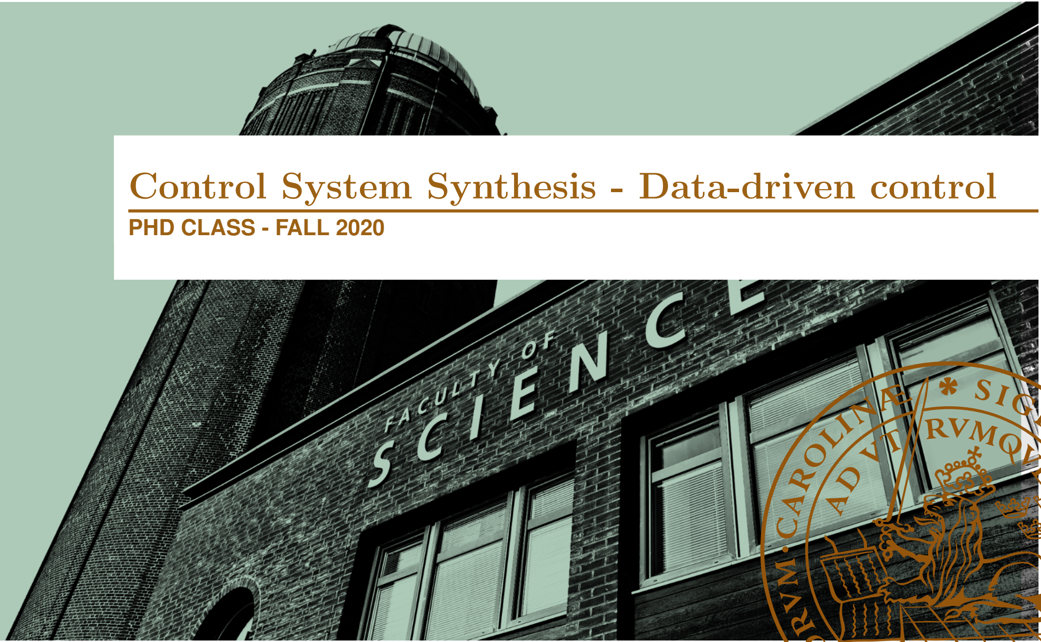
Lund University

Control System Synthesis - Data-driven control

PHD CLASS - FALL 2020



1. Introduction
2. Fundamentals
3. Design techniques
   - PID control
   - Optimal control and LQG
   - Robust control
   - Model Predictive Control
   - Adaptive Control
   - **Data-driven Control**

Content overview

Introduction to data-driven control

The importance of data-driven approaches

- “Modern” control theory requires a description of the system and of the disturbances

Introduction to data-driven control

The importance of data-driven approaches

- “Modern” control theory requires a description of the system and of the disturbances

→ simply not available in some practical problems

Introduction to data-driven control

The importance of data-driven approaches

- “Modern” control theory requires a description of the system and of the disturbances

→ simply not available in some practical problems

→ Motivating example: large-scale industrial processes

Introduction to data-driven control

The importance of data-driven approaches

- "Modern" control theory requires a description of the system and of the disturbances

$\rightarrow$ simply not available in some practical problems

$\rightarrow$ Motivating example: large-scale industrial processes

- **Identification** $\rightarrow$ research in this area since the 1960s

Introduction to data-driven control

The importance of data-driven approaches

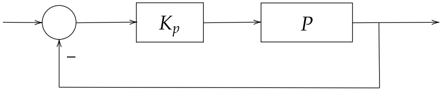
- “Modern” control theory requires a description of the system and of the disturbances

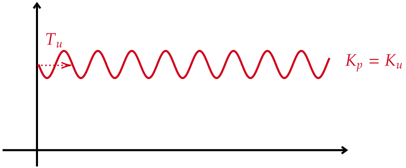
$\rightarrow$ simply not available in some practical problems

$\rightarrow$ Motivating example: large-scale industrial processes

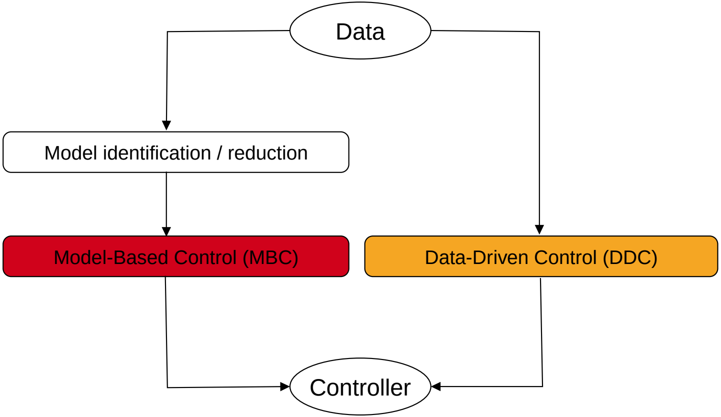
- **Identification** $\rightarrow$ research in this area since the 1960s
- **Autotuning**

- Ziegler-Nichols formulas for PIDs

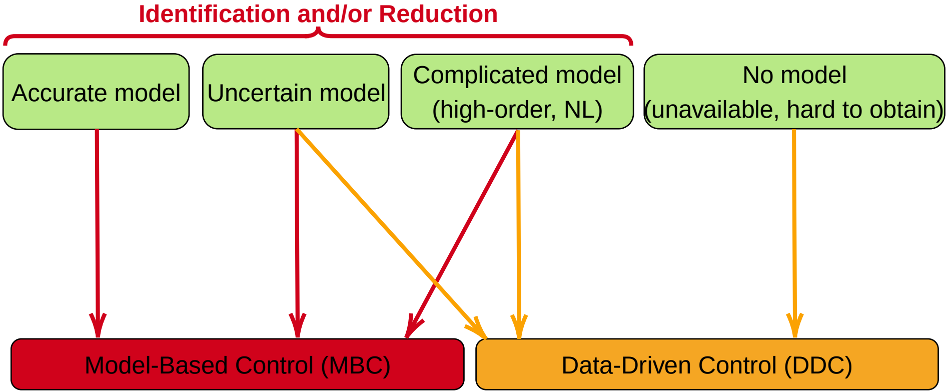




- PID autotuning (Karl Johan and Tore)



Data
Model identification / reduction
Model-Based Control (MBC)
Data-Driven Control (DDC)
Controller


Identification and/or Reduction
Accurate model
Uncertain model
Complicated model (high-order, NL)
No model (unavailable, hard to obtain)
Model-Based Control (MBC)
Data-Driven Control (DDC)

- What are good data? (experiment design)

- What are good data? (experiment design)

About MBC

- When is a model ”too hard to obtain”?
- What is a good model?
- What is the relation between identification and control?

- What are good data? (experiment design)

About MBC

- When is a model ”too hard to obtain”?
- What is a good model?
- What is the relation between identification and control?

→ stakes of the application

→ purpose of the model

→ accuracy vs complexity trade-off

Introduction to data-driven control

Model-based and data-driven control

- What are good data? (experiment design)

About MBC

- When is a model ”too hard to obtain”?
- What is a good model?
- What is the relation between identification and control?

→ stakes of the application

→ purpose of the model

→ accuracy vs complexity trade-off

About DDC

- How does DDC handle uncertainty and complexity?
- To what extent a DDC technique is model-free?

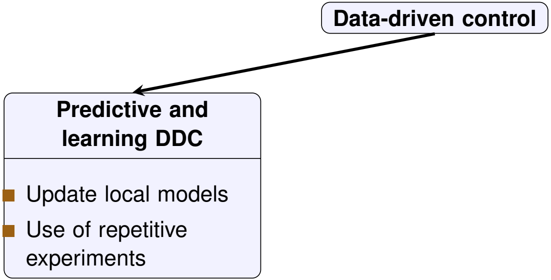
"Control theory makes no claims about the performance or stability of physical systems; only about their models."

"[Model-based control] starts and ends with the model. To some extent, it may be called model theory rather than control theory."

1. *Controller validation*, Brozenec, T. F., Tsao, T. C., Safonov, M. G. (2001), in *International Journal of Adaptive Control and Signal Processing*.
2. *From model-based control to data-driven control: Survey, classification and perspective*, Hou, Z. S., Wang, Z. (2013), in *Information Sciences*.

Introduction to data-driven control

Overview of data-driven control technique



Introduction to data-driven control

Overview of data-driven control technique

Data-driven control

Predictive and learning DDC

- Update local models
- Use of repetitive experiments

Robust DDC

$$\begin{aligned} \|W_1 S\|_\infty &\leq 1 \\ \|W_2 KS\|_\infty &\leq 1 \end{aligned}$$

Introduction to data-driven control

Overview of data-driven control technique

Data-driven control

Predictive and learning DDC

- Update local models
- Use of repetitive experiments

Robust DDC

$$\begin{aligned} \|\mathbf{W_1S}\|_\infty &\leq 1 \\ \|\mathbf{W_2KS}\|_\infty &\leq 1 \end{aligned}$$

Model-reference DDC

$$\min_{\mathbf{K}} \left\|(1+\mathbf{PK})^{-1}\mathbf{PK}-\mathbf{M}\right\|$$

Introduction to data-driven control

Overview of data-driven control technique

Data-driven control

Predictive and learning DDC
- Update local models
- Use of repetitive experiments

Robust DDC

$$\begin{aligned} \|\mathbf{W_1 S}\|_\infty &\leq 1 \\ \|\mathbf{W_2 KS}\|_\infty &\leq 1 \end{aligned}$$

Model-reference DDC

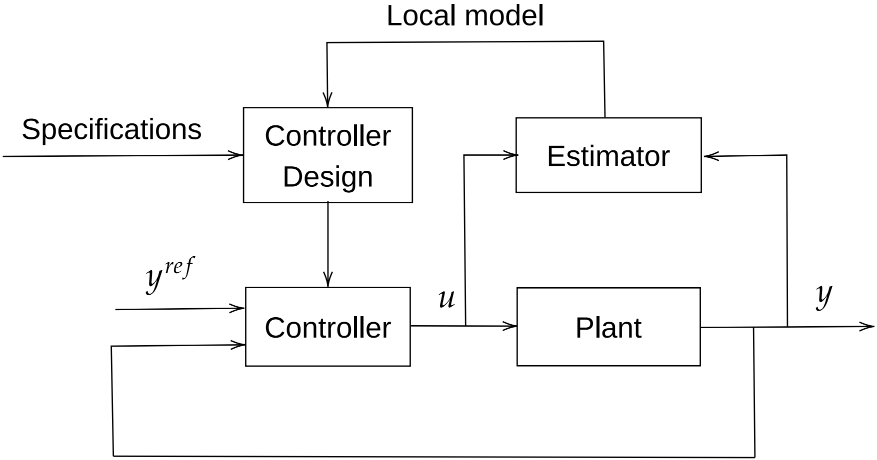
$$\min_{\mathbf{K}} \left\| (1 + \mathbf{PK})^{-1}\mathbf{PK} - \mathbf{M} \right\|$$

Other classifications:
- Online/offline data
- Controller structure
- System knowledge

Predictive and learning DDC

Predictive and learning DDC

Use of local models



Self-Tuning Regulators (STR)

- Step 1: Least squares parameter estimation

$$y(t) + \sum_{i=1}^{m} \alpha_i y(t-k-i) = \beta_0 \left(u(t-k-1) + \sum_{i=1}^{l} \beta_i u(t-k-i-1) \right)$$

- Step 2: Control

$$u(t) = \frac{z^{l-m+1}\alpha(z)}{\beta(z)} y(t)$$

1 *On self tuning regulators*, Åström, K. J., Wittenmark, B. (1973), in *Automatica*.

Predictive and learning DDC

Use of local models

Self-Tuning Regulators (STR)

- Step 1: Least squares parameter estimation

$$y(t) + \sum_{i=1}^{m} \alpha_i y(t-k-i)$$
$$= \beta_0 \left(u(t-k-1) + \sum_{i=1}^{l} \beta_i u(t-k-i-1) \right)$$

- Step 2: Control

$$u(t) = \frac{z^{l-m+1}\alpha(z)}{\beta(z)} y(t)$$

Model-Free Control or iPIDs

- Step 1: Ultra local estimation

$$y^{(\nu)} = F + \alpha u$$

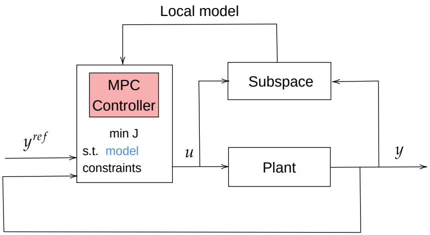
- Step 2: Control

$$u = -\frac{F - y^{(\nu)} + C(z)e}{\alpha}$$

1 *On self tuning regulators*, Åström, K. J., Wittenmark, B. (1973), in *Automatica*.
2 *Model-free control*, Fliess, M., Join, C. (2013), in *International Journal of Control*.

Predictive and learning DDC

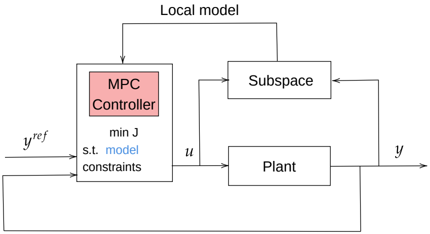
Data-driven MPC



1 *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*, Huang, B., Kadali, R. (2008), Springer.

Predictive and learning DDC

Data-driven MPC



Subspace approach

- Prediction model:

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

- Subspace id: Find (A, B, C, D) from IO data $\{u_t\}_{t=1}^N$ and $\{y_t\}_{t=1}^N$

1 *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*, Huang, B., Kadali, R. (2008), Springer.

Subspace algorithm

1. Build data matrices $\mathcal{Y}$ and $\mathcal{U}$ from the data

$$\mathcal{Y} = \mathcal{O}_n \mathcal{X} + \Gamma \mathcal{U}$$

Subspace algorithm

1. Build data matrices $\mathcal{Y}$ and $\mathcal{U}$ from the data

$$\mathcal{Y} = \mathcal{O}_n \mathcal{X} + \Gamma \mathcal{U}$$

2. Project to estimate the rank space of $\mathcal{O}_n$

$$\mathcal{Y}\mathcal{U}^{\perp} = \mathcal{O}_n \mathcal{X}\mathcal{U}^{\perp}$$

Subspace algorithm

1. Build data matrices $\mathcal{Y}$ and $\mathcal{U}$ from the data

$$\mathcal{Y} = \mathcal{O}_n \mathcal{X} + \Gamma \mathcal{U}$$

2. Project to estimate the rank space of $\mathcal{O}_n$

$$\mathcal{Y}\mathcal{U}^{\perp} = \mathcal{O}_n \mathcal{X} \mathcal{U}^{\perp}$$

3. Get the matrices A and C

Subspace algorithm

1. Build data matrices $\mathcal{Y}$ and $\mathcal{U}$ from the data

$$\mathcal{Y} = \mathcal{O}_n \mathcal{X} + \Gamma \mathcal{U}$$

2. Project to estimate the rank space of $\mathcal{O}_n$

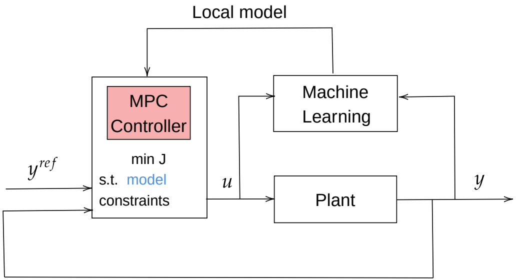
$$\mathcal{Y}\mathcal{U}^{\perp} = \mathcal{O}_n \mathcal{X}\mathcal{U}^{\perp}$$

3. Get the matrices A and C
4. Get the matrices B and D from the data (least squares)

Predictive and learning DDC

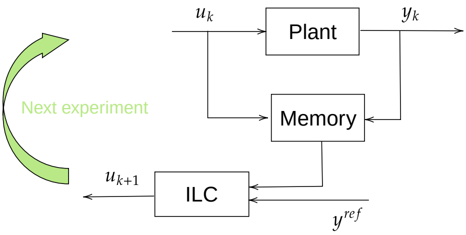
Use of local models

Many other ways to update a local model from data



- *Learning-based model predictive control: Toward safe learning in control*, Hewing, L., Wabersich, K. P., Menner, M., Zeilinger, M. N. in *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.

Iterative Learning Control



$$y_k \underset{k\to\infty}{\longrightarrow} y^{ref}$$

$$u_{k+1}(z) = u_k(z) + L(z)(y^{ref}(z) - y_k(z))$$

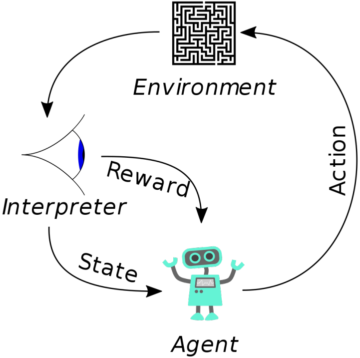
$$e_{k+1}(z) = (1 - P(z)L(z))e_k(z)$$

- *A survey of iterative learning control: A learning-based method for high-performance tracking control*, D. Bristow, M. Tharayil, and A. Alleyne, 2006, in *IEEE Control Systems Magazine*.
- *Iterative learning control: Analysis, design, and experiments*, M. Norrlof, Ph.D. dissertation, University of Linkoping, 2000.

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

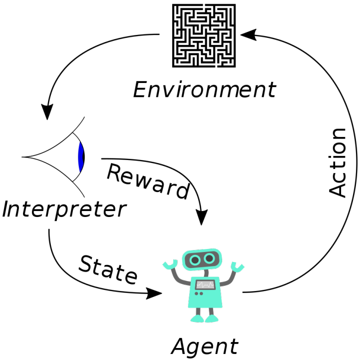


- **Markov Decision process**

Predictive and learning DDC

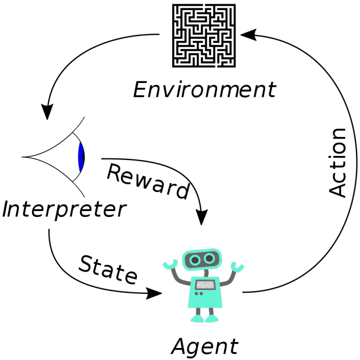
Use of repetitive experiments

Reinforcement learning



- **Markov Decision process**
- set of environment and agent states

Reinforcement learning

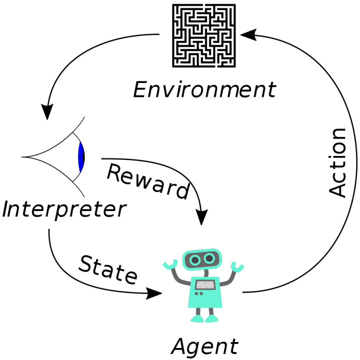


- **Markov Decision process**
- set of environment and agent states
- set of actions of the agent

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

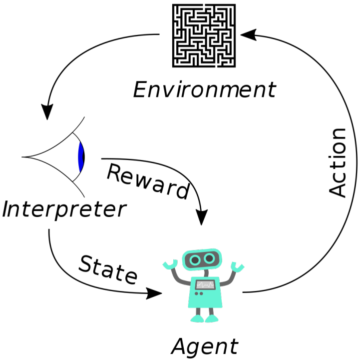


- **Markov Decision process**
- set of environment and agent states
- set of actions of the agent
- probability of transition at time t from state s to state s' under action a

Predictive and learning DDC

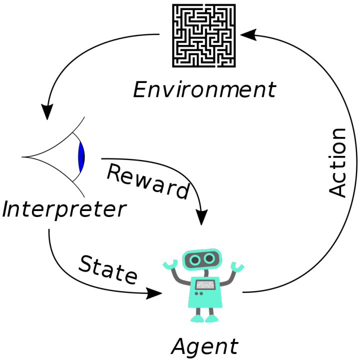
Use of repetitive experiments

Reinforcement learning



- **Markov Decision process**
- set of environment and agent states
- set of actions of the agent
- probability of transition at time t from state s to state s' under action a
- reward after transition from s to s' under action a

Reinforcement learning



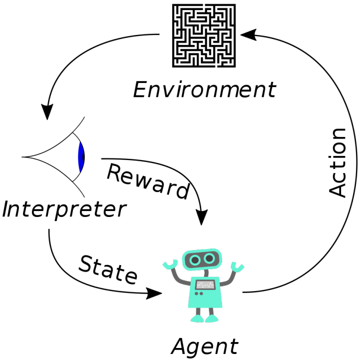
- **Markov Decision process**
- set of environment and agent states
- set of actions of the agent
- probability of transition at time t from state s to state s' under action a
- reward after transition from s to s' under action a

→ **Objective**: learn a policy which maximizes the expected cumulative reward

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning



- **Markov Decision process**
- set of environment and agent states
- set of actions of the agent
- probability of transition at time t from state s to state s' under action a
- reward after transition from s to s' under action a

$\rightarrow$ **Objective**: learn a policy which maximizes the expected cumulative reward

- comparison with an optimal agent $\rightarrow$ regret
- Applications: robotics, flight dynamics, multi-agents, planning (model-based)

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning
  1. $Q(s, a)$ is a cost function, initialized by the user

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning
  1. $Q(s, a)$ is a cost function, initialized by the user
  2. The action to take is given by $\pi(s) = \max_a Q_i(s, a)$

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning
  1. $Q(s, a)$ is a cost function, initialized by the user
  2. The action to take is given by $\pi(s) = \max_a Q_i(s, a)$
  3. Q is updated at each step $s \xrightarrow[a]{} s'$ giving reward r

$$Q_{i+1}(s, a) = (1 - \alpha) Q_i(s, a) + \alpha \left(r + \gamma \max_{a'} Q_i(s', a') \right)$$

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning
  1. $Q(s, a)$ is a cost function, initialized by the user
  2. The action to take is given by $\pi(s) = \max_a Q_i(s, a)$
  3. Q is updated at each step $s \xrightarrow[a]{} s'$ giving reward r

$$Q_{i+1}(s, a) = (1-\alpha)Q_i(s, a) + \alpha \left(r + \gamma \max_{a'} Q_i(s', a') \right)$$

- α is the **learning rate**

$\rightarrow$ $\alpha = 0 \rightarrow$ learn nothing, exploitation

$\rightarrow$ $\alpha = 1 \rightarrow$ ignore prior knowledge, only exploration

Predictive and learning DDC

Use of repetitive experiments

Reinforcement learning

- **Data-driven case:** the only way to collect information about the environment is to interact with it
- **Exploration vs Exploitation**
- A data-driven technique: Q-learning
  1. $Q(s, a)$ is a cost function, initialized by the user
  2. The action to take is given by $\pi(s) = \max_a Q_i(s, a)$
  3. Q is updated at each step $s \xrightarrow[a]{} s'$ giving reward r

$$Q_{i+1}(s, a) = (1 - \alpha)Q_i(s, a) + \alpha \left(r + \gamma \max_{a'} Q_i(s', a') \right)$$

- α is the **learning rate**
- γ is the **discount factor** (importance of future reward)

$\rightarrow$ $\gamma = 0 \rightarrow$ short-term

$\rightarrow$ $\gamma = 1 \rightarrow$ long-term

Robust DDC

Robust DDC

Using convex optimization

$$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \dots m\} \qquad K(s,\rho) = \rho\phi(s)$$

- Classical robust requirement:
  $\| |W_1 S| + |W_2 T| \|_\infty < 1$

- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

Robust DDC

Using convex optimization

$$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \dots m\}$$

$$K(s, \rho) = \rho\phi(s)$$

- Classical robust requirement:
  $\| |W_1 S| + |W_2 T| \|_\infty < 1$
- Data-driven version:
  $\| |W_1(\jmath\omega)S(\jmath\omega)| + |W_2(\jmath\omega)T(\jmath\omega)| \|_\infty < 1$

- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

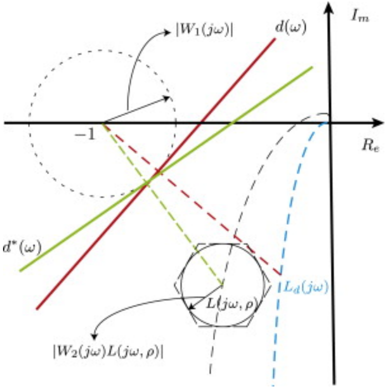
Robust DDC

Using convex optimization

$$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \dots m\}$$

$$K(s, \rho) = \rho\phi(s)$$

- Classical robust requirement: $\||W_1 S| + |W_2 T|\|_\infty < 1$
- Data-driven version: $\||W_1(\jmath\omega)S(\jmath\omega)| + |W_2(\jmath\omega)T(\jmath\omega)|\|_\infty < 1$
- Constraints on the Nyquist plot: $\||W_1(\jmath\omega)| + |W_2(\jmath\omega)L(\rho, \jmath\omega)|\|_\infty < |1 + L(\rho, \jmath\omega)|$



- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

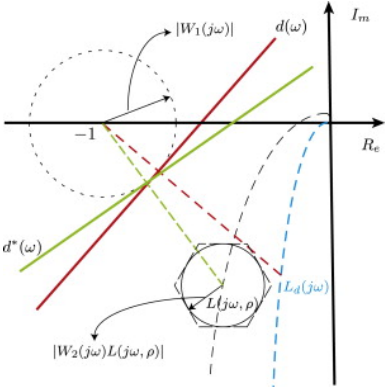
Robust DDC

Using convex optimization

$$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \dots m\} \qquad K(s,\rho) = \rho\phi(s)$$

- Classical robust requirement: $\| |W_1 S| + |W_2 T| \|_\infty < 1$
- Data-driven version: $\| |W_1(\jmath\omega)S(\jmath\omega)| + |W_2(\jmath\omega)T(\jmath\omega)| \|_\infty < 1$
- Constraints on the Nyquist plot: $\| |W_1(\jmath\omega)| + |W_2(\jmath\omega)L(\rho,\jmath\omega)| \|_\infty < |1 + L(\rho,\jmath\omega)|$

$\rightarrow$ requires a desirable open-loop $L_d \approx L(\rho)$



- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

Robust DDC

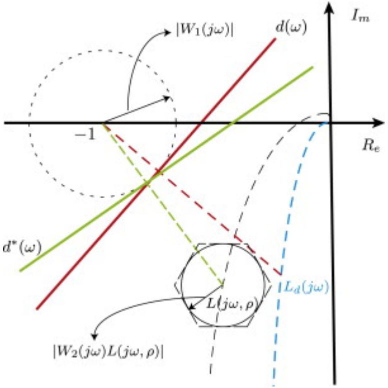
Using convex optimization

$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \dots m\}$ $\qquad K(s, \rho) = \rho\phi(s)$

- Classical robust requirement: $\||W_1 S| + |W_2 T|\|_\infty < 1$
- Data-driven version: $\||W_1(\jmath\omega)S(\jmath\omega)| + |W_2(\jmath\omega)T(\jmath\omega)|\|_\infty < 1$
- Constraints on the Nyquist plot: $\||W_1(\jmath\omega)| + |W_2(\jmath\omega)L(\rho, \jmath\omega)|\|_\infty < |1 + L(\rho, \jmath\omega)|$

$\rightarrow$ requires a desirable open-loop $L_d \approx L(\rho)$

$\rightarrow$ for SISO systems only



- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

Robust DDC

Using convex optimization

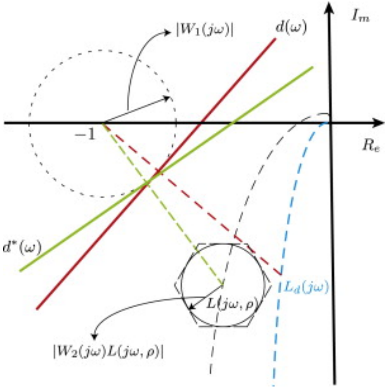
$$\{G(\jmath\omega_i)(1 + W_i(\jmath\omega_i)\Delta),\ i = 1 \ldots m\} \qquad K(s,\rho) = \rho\phi(s)$$

- Classical robust requirement:
  $\| |W_1 S| + |W_2 T| \|_\infty < 1$
- Data-driven version:
  $\| |W_1(\jmath\omega)S(\jmath\omega)| + |W_2(\jmath\omega)T(\jmath\omega)| \|_\infty < 1$
- Constraints on the Nyquist plot:
  $\| |W_1(\jmath\omega)| + |W_2(\jmath\omega)L(\rho,\jmath\omega)| \|_\infty <$
  $|1 + L(\rho,\jmath\omega)|$

$\rightarrow$ requires a desirable open-loop $L_d \approx L(\rho)$

$\rightarrow$ for SISO systems only

$\rightarrow$ can be solved with convex optimisation



- *Fixed-order $\mathcal{H}_\infty$ controller design for nonparametric models by convex optimization*, Karimi, A., Galdos, G. (2010) in *Automatica*.

Robust DDC

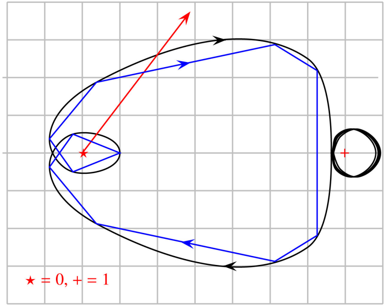
Using non-convex optimization

$$\begin{aligned} \text{minimize} \quad & \max_{i=1\ldots N} \overline{\sigma}(T_{zw}(\jmath\omega_i)) \\ \text{subject to} \quad & K \in \mathcal{K} \text{ stabilizing} \end{aligned}$$

1. Initialization with a stabilizing controller
2. Find a polyhedral model
3. Primary descent (trust-region)
4. Nyquist stability test

→ satisfied: iterate

→ not satisfied: redo the descent with a smaller trust radius



- *Structured $\mathcal{H}_\infty$-control of infinite-dimensional systems*, Apkarian, P., Noll, D. (2018) in *International Journal of Robust and Nonlinear Control*.

Model reference DDC

Model reference DDC

Structured controller

$$\min_{\theta} J(\theta) = \left\| \frac{\mathbf{P}(z)\mathbf{K}(z,\theta)}{1+\mathbf{P}(z)\mathbf{K}(z,\theta)} - \mathbf{M}(z) \right\|_2^2$$

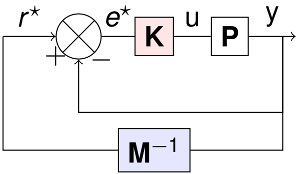
Model reference DDC

Structured controller

$$\min_{\theta} J(\theta) = \left\| \frac{\mathbf{P}(z)\mathbf{K}(z,\theta)}{1+\mathbf{P}(z)\mathbf{K}(z,\theta)} - \mathbf{M}(z) \right\|_2^2$$

VRFT

$$J_{VRFT}^{N}(\theta) = \frac{1}{N}\sum_{k=1}^{N}\left(u_F(t_k) - \mathbf{K}(z,\theta)e_F^{\star}(t_k)\right)^2$$

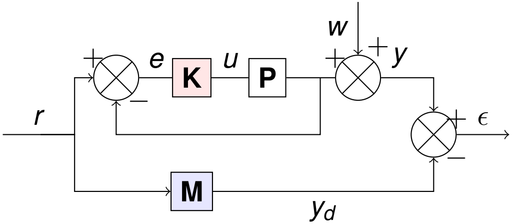


- *Virtual reference feedback tuning: a direct method for the design of feedback controllers*, Campi, M. C., Lecchini, A., Savaresi, S. M. (2002) in *Automatica*.

Model reference DDC

Structured controller

$$\min_{\theta} J(\theta) = \left\| \frac{\mathbf{P}(z)\mathbf{K}(z,\theta)}{1+\mathbf{P}(z)\mathbf{K}(z,\theta)} - \mathbf{M}(z) \right\|_2^2$$



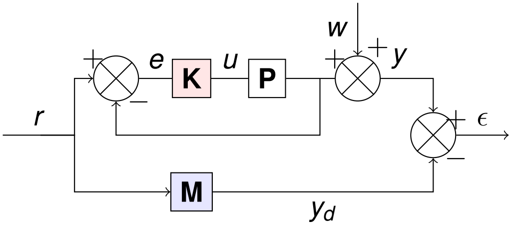
$$\epsilon(s) = \left(\mathbf{M}(s) - \frac{\mathbf{P}(s)\mathbf{K}(\theta,s)}{1+\mathbf{P}(s)\mathbf{K}(\theta,s)} \right) r(s) + \frac{1}{1+\mathbf{P}(s)\mathbf{K}(\theta,s)} w(s)$$

- *Correlation-based tuning of decoupling multivariable controllers*, Mišković, L., Karimi, A., Bonvin, D., Gevers, M. (2007) in *Automatica*.

Model reference DDC

Structured controller

$$\min_{\theta} J(\theta) = \left\| \frac{\mathbf{P}(z)\mathbf{K}(z,\theta)}{1+\mathbf{P}(z)\mathbf{K}(z,\theta)} - \mathbf{M}(z) \right\|_2^2$$



$$\epsilon(s) = \left(\mathbf{M}(s) - \frac{\mathbf{P}(s)\mathbf{K}(\theta,s)}{1+\mathbf{P}(s)\mathbf{K}(\theta,s)} \right) r(s) + \frac{1}{1+\mathbf{P}(s)\mathbf{K}(\theta,s)} w(s)$$

$$\frac{\partial \epsilon(\theta)}{\partial \theta} = \frac{1}{\mathbf{K}(\theta)} \frac{\partial \mathbf{K}(\theta)}{\partial \theta} \left(\frac{\mathbf{PK}(\theta)}{1+\mathbf{PK}(\theta)} (r - y(\theta)) \right)$$

Model reference DDC

Structured controller

$$\min_{\theta} J(\theta) = \left\| \frac{\mathbf{P}(z)\mathbf{K}(z,\theta)}{1+\mathbf{P}(z)\mathbf{K}(z,\theta)} - \mathbf{M}(z) \right\|_2^2$$

Key assumptions:

- The ideal controller $K^\star$ belongs to the chosen class $\mathcal{K}$ of controllers
- The reference model is achievable by the plant

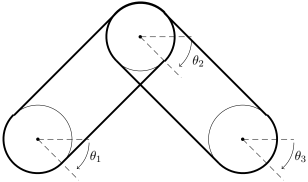
$$M(z) = 0$$

$$M(p) = 1$$

- **Hard to meet in a data-driven framework**

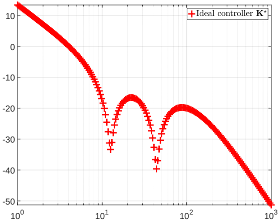
Model reference DDC

Unstructured controller

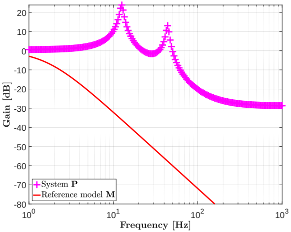


$\rightarrow$ 200 points logarithmically spaced between 1 and 10^3rad.s^{-1}

$$\mathbf{M}(s) = \frac{1}{\frac{1}{\omega_0^2}s^2 + \frac{2\xi}{\omega_0}s + 1} \quad \omega_0 = 10\text{rad.s}^{-1},\ \xi = 1$$



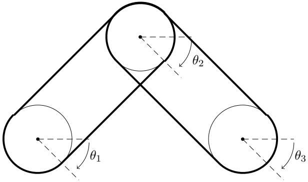
(a) Controller identification



(b) Resulting closed-loops

Model reference DDC

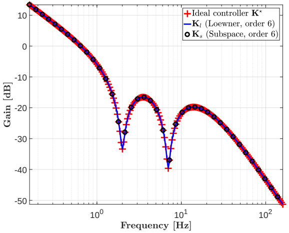
Unstructured controller



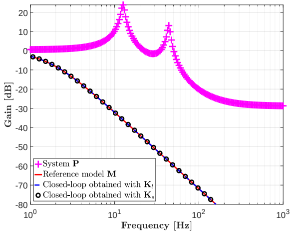
$\rightarrow$ 200 points logarithmically spaced between 1 and 10^3rad.s^{-1}

$$\mathbf{M}(s) = \frac{1}{\frac{1}{\omega_0^2}s^2 + \frac{2\xi}{\omega_0}s + 1} \quad \omega_0 = 10\text{rad.s}^{-1},\ \xi = 1$$

$$\mathbf{K}^\star(\imath\omega_i) = \Phi_i^{-1}\mathbf{M}(\imath\omega_i)(I - \mathbf{M}(\imath\omega_i))^{-1}$$



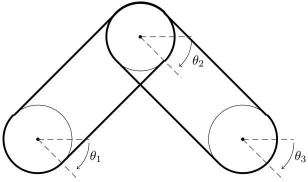
(c) Controller identification



(d) Resulting closed-loops

Model reference DDC

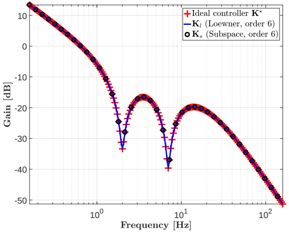
Unstructured controller



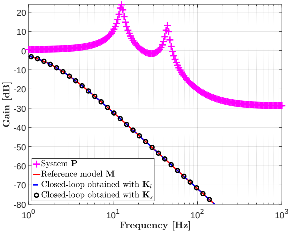
$$\mathbf{P}(s) = \frac{0.03616(s-140.5)(s-40)^3}{(s^2+1.071s+157.9)(s^2+3.172s+1936)}$$

$$\mathbf{M}(s) = \frac{1}{\frac{1}{\omega_0^2}s^2+\frac{2\xi}{\omega_0}s+1} \quad \omega_0 = 10\text{rad.s}^{-1},\ \xi = 1$$

$$\mathbf{K}^\star(s) = k\frac{(s^2+1.071s+157.9)(s^2+3.172s+1936)}{s(s+20)(s-140.5)(s-40)^3}$$



(e) Controller identification

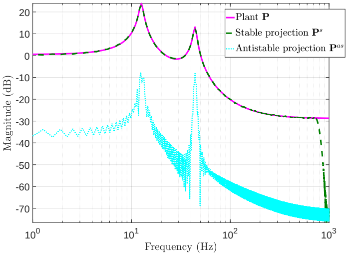


(f) Resulting closed-loops

Model reference DDC
Unstructured controller

Step 1: Determine the plant's limitations from its data

$$\mathbf{P} = \mathbf{P}^s + \mathbf{P}^{as}$$



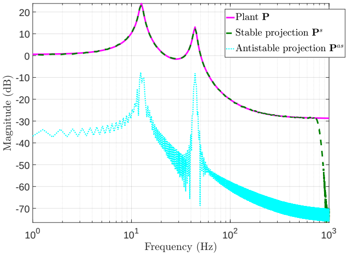
Projection of $\{\omega_i, \Phi_i\}_{i=1}^N$: the system is stable.

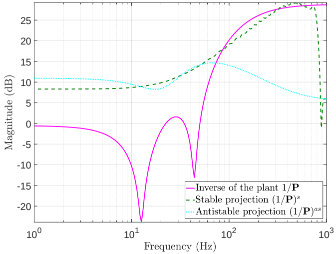
Model reference DDC

Unstructured controller

Step 1: Determine the plant's limitations from its data

$$\mathbf{P} = \mathbf{P}^s + \mathbf{P}^{as} \qquad \frac{1}{\mathbf{P}} = \frac{1}{\mathbf{P}}^{s} + \frac{1}{\mathbf{P}}^{as}$$





Projection of $\{\omega_i, \Phi_i\}_{i=1}^N$: the system is stable.

Projection of $\{\omega_i, \Phi_i^{-1}\}_{i=1}^N$: the system is NMP.

Model reference DDC

Unstructured controller

Step 1: Determine the plant's limitations from its data

$$\frac{1}{\mathbf{P}} = \frac{1}{\mathbf{P}}^{s} + \frac{1}{\mathbf{P}}^{as}$$

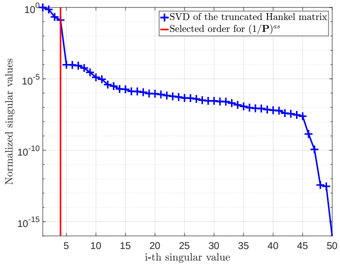
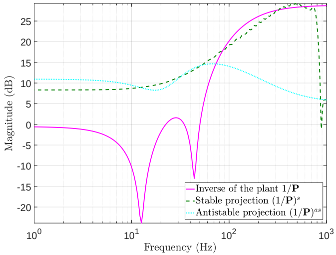


Figure: Determination of the number of RHP zeros from Principal Hankel Component analysis.



Projection of $\{\omega_i, \Phi_i^{-1}\}_{i=1}^{N}$: the system is NMP.

Step 1: Determine the plant's limitations from its data

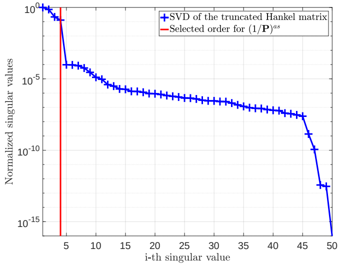


Figure: Determination of the number of RHP zeros from Principal Hankel Component analysis.

System's RHP zeros	Estimation
140.5	140.49
40	39.32
40	39.32
40	41.4

Model reference DDC

Unstructured controller

Step 1: Determine the plant's limitations from its data

$$\mathbf{M}_f = \mathbf{M}\mathbf{B}_z$$

$$\mathbf{B}_z(s) = \prod_{i=1}^{n_z} \frac{s - z_i}{s + z_i}$$

$$|\mathbf{B}_z(\imath\omega)| = 1$$

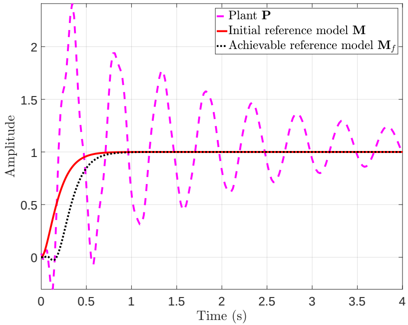
System's RHP zeros	Estimation
140.5	140.49
40	39.32
40	39.32
40	41.4

Step 1: Determine the plant's limitations from its data

$$\mathbf{M}_f = \mathbf{M}\mathbf{B}_z$$

$$\mathbf{B}_z(s) = \prod_{i=1}^{n_z} \frac{s - z_i}{s + z_i}$$

$$|\mathbf{B}_z(\imath\omega)| = 1$$



Model reference DDC

Unstructured controller

Step 2: Controller identification

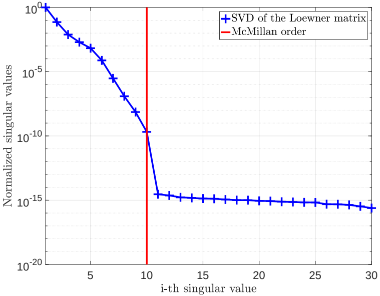


Figure: Singular Value Decomposition of the Loewner matrix to determine the McMillan order.

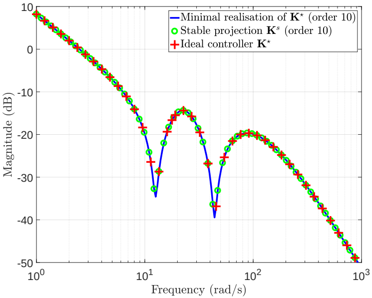


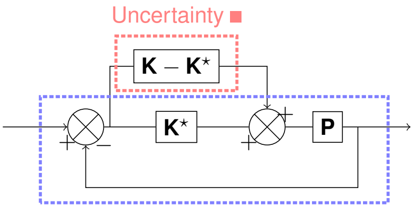
Figure: Projection on $\mathcal{RH}_\infty$ to enforce stability of $\mathbf{K}$.

Model reference DDC

Unstructured controller

Step 3: Controller reduction under stability constraint

Application of the small-gain theorem



The resulting closed-loop is well-posed and internally stable for all stable Δ such that $\|\Delta\|_\infty \leq \beta$ if and only if $\|(1 - \mathbf{M}_f)\mathbf{P}\|_\infty < \frac{1}{\beta}$.

$\rightarrow$ Limiting the controller modelling error allows to ensure closed-loop internal stability!

Step 3: Controller reduction under stability constraint

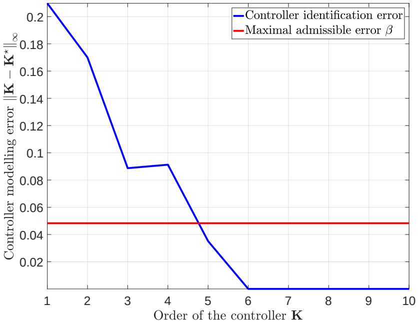


Figure: Evolution of the controller modelling error.

The resulting closed-loop is well-posed and internally stable for all stable Δ such that $\|\Delta\|_\infty \leq \beta$ if and only if $\|(1-\mathbf{M}_f)\mathbf{P}\|_\infty < \frac{1}{\beta}$.

$\rightarrow$ Limiting the controller modelling error allows to ensure closed-loop internal stability!

Step 3: Controller reduction under stability constraint

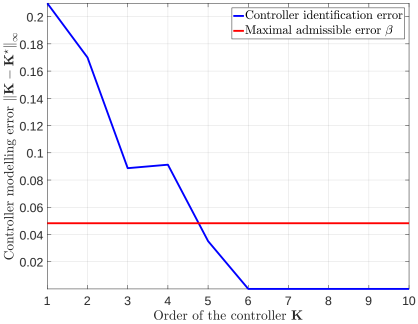


Figure: Evolution of the controller modelling error.

Model reference DDC

Unstructured controller

Taking a slower reference model: $\omega_0 = 5\text{rad.s}^{-1}$ ($\omega_0 = 10\text{rad.s}^{-1}$ until now)

$$\mathbf{M}_2(s) = \frac{1}{0.04s^2+0.4s+1} \qquad \mathbf{M}_{f2} = \mathbf{M}_2\mathbf{B}_z \qquad \beta = 0.0609$$

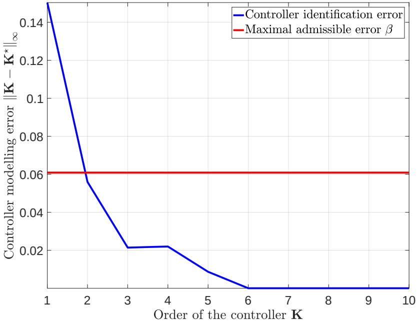


Figure: Evolution of the controller modelling error for relaxed specifications.

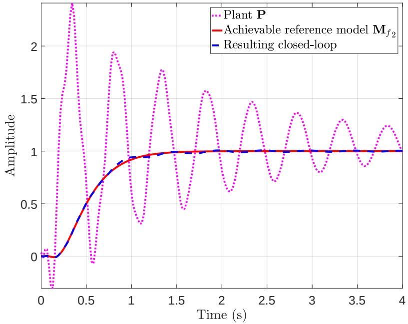


Figure: Time-domain simulation.

Conclusion

- **Knowledge of the plant**

Conclusion

- **Knowledge of the plant**
  - Predictive and learning DDC generates a lot of knowledge about the system

Conclusion

- **Knowledge of the plant**
  - Predictive and learning DDC generates a lot of knowledge about the system
  - Robust DDC and model reference DDC requires some insight about the system (achievable specifications or initial stabilizing controller)

Conclusion

- **Knowledge of the plant**
  - Predictive and learning DDC generates a lot of knowledge about the system
  - Robust DDC and model reference DDC requires some insight about the system (achievable specifications or initial stabilizing controller)
- **Controller structure** may be hard to choose when needed

Conclusion

- **Knowledge of the plant**
  - Predictive and learning DDC generates a lot of knowledge about the system
  - Robust DDC and model reference DDC requires some insight about the system (achievable specifications or initial stabilizing controller)
- **Controller structure** may be hard to choose when needed
- **Online or offline**

LUND UNIVERSITY

Conclusion

- **Knowledge of the plant**
  - Predictive and learning DDC generates a lot of knowledge about the system
  - Robust DDC and model reference DDC requires some insight about the system (achievable specifications or initial stabilizing controller)
- **Controller structure** may be hard to choose when needed
- **Online or offline**
- Different objective formulations

Conclusion

Things we have not talked about

- Control of and over networks

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)
- Going towards other areas (biology, medicine, economics, social sciences)

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)
- Going towards other areas (biology, medicine, economics, social sciences)
- Much more to do with learning

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)
- Going towards other areas (biology, medicine, economics, social sciences)
- Much more to do with learning

→ Theory and applications are strongly linked!

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)
- Going towards other areas (biology, medicine, economics, social sciences)
- Much more to do with learning

→ Theory and applications are strongly linked!

Conclusion

Things we have not talked about

- Control of and over networks
- Autonomous systems (cars, robots)
- Interplay between computing and control
- Tackling complexity (size, interactions, nonlinearities,...)
- Going towards other areas (biology, medicine, economics, social sciences)
- Much more to do with learning

$\rightarrow$ Theory and applications are strongly linked!

Don't forget the bigger picture!