Control-oriented modelling - what is it?

Damiano Varagnolo

October 2018

Today's presentation

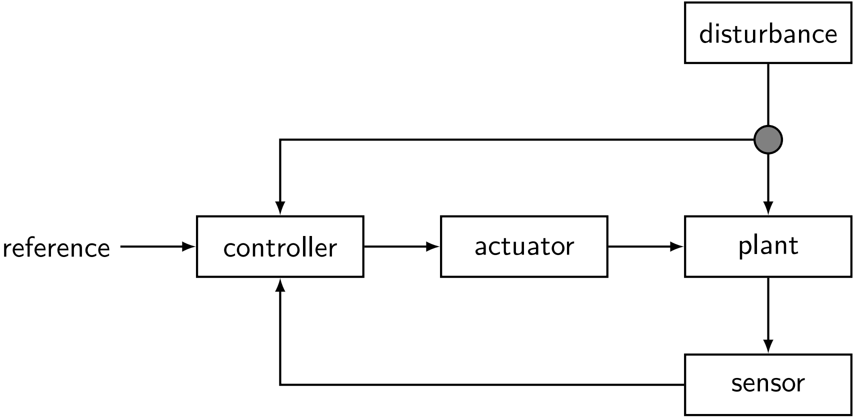
aim: what types of models can we use to operate a system, and how can we obtain them?

path:

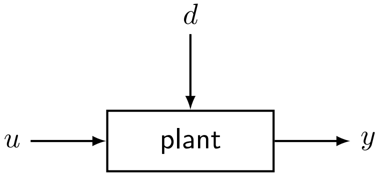
1. what is control?
2. what are control-oriented models?
3. how can we get control-oriented models from field measurements?

introducing today’s ingredients

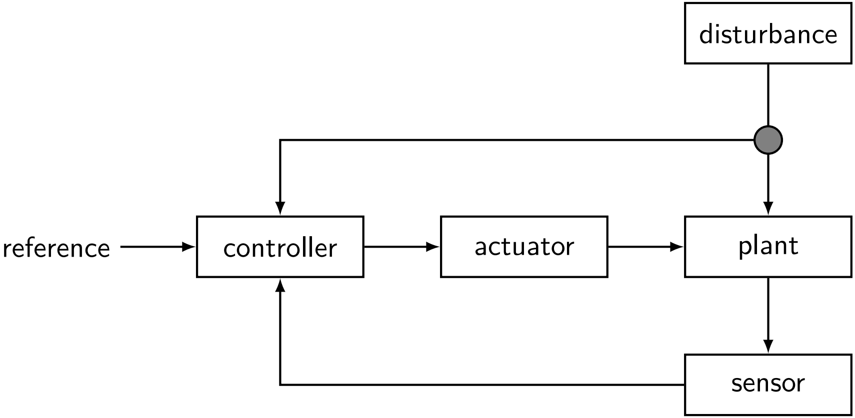
Controller types - Feedforward-Feedback



What is a model?



What is a control-oriented model?



How do we represent a control-oriented model?

Definition of state space representation: *set of first-order differential equations among a finite set of inputs, outputs and state variables satisfying the separation principle, i.e., the future output depends only on the current state and the future input*

How do we represent a control-oriented model?

Definition of state space representation: *set of first-order differential equations among a finite set of inputs, outputs and state variables satisfying the separation principle, i.e., the future output depends only on the current state and the future input*

Implication: the state summarizes the effect of past inputs on future output

(sort of a "memory" of the system)

State space representations - Example

Rechargeable flashlight:

- input $u =$ on / off button
- state $x =$ level of charge of the battery
- output $y =$ how much light is emitted

State space representations

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\ ;\ \theta\right) \\
\boldsymbol{y} &= \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}\ ;\ \theta\right)
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{x}(k+1) &= \boldsymbol{f}\big(\boldsymbol{x}(k), \boldsymbol{u}(k)\ ;\ \theta\big) \\
\boldsymbol{y}(k) &= \boldsymbol{g}\big(\boldsymbol{x}(k), \boldsymbol{u}(k)\ ;\ \theta\big)
\end{aligned}
$$

Definition: linear systems

$$\boldsymbol{u} \longrightarrow \boxed{G} \longrightarrow \boldsymbol{y}$$

Definition (linearity)

$G(\cdot)$ *is linear iff* $\forall \alpha_1, \alpha_2, \boldsymbol{u}_1, \boldsymbol{u}_2$

$$\boldsymbol{y} = G\left(\alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2\right) = \alpha_1 G\left(\boldsymbol{u}_1\right) + \alpha_2 G\left(\boldsymbol{u}_2\right) = \alpha_1 \boldsymbol{y}_1 + \alpha_2 \boldsymbol{y}_2$$

Definition: nonlinear systems

anything that is not linear

Linear vs. nonlinear state-space systems

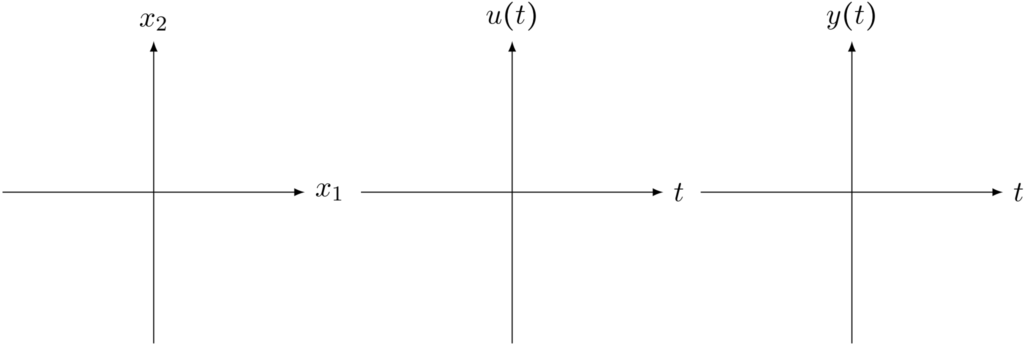
$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} \end{aligned} \qquad\qquad \begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\,;\ \theta\right) \\ \boldsymbol{y} &= \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}\,;\ \theta\right) \end{aligned}$$

how can we do control?

Control with linear models (LQR)

$$\begin{cases} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + Bu \\ y &= C\boldsymbol{x} \end{cases}$$

Idea: $$J\left(y,u\right) = \rho \left\|y\right\|_2^2 + \left\|u\right\|_2^2 \qquad \left\|\chi\right\|_2^2 := \int_0^{+\infty} \chi^2(t)dt$$



Control with linear models (LQR) - fundamental result

under the simplifying assumption that the systems that we consider are fully controllable

Theorem

If

$$\begin{cases} \dot{\boldsymbol{x}} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} \end{cases} \qquad J(y, u) = \rho \|y\|_2^2 + \|u\|_2^2$$

then

$$\arg\min_{u \in \mathbb{R}_u} J(y, u) = -K\boldsymbol{x}$$

for an opportune K.

Control with linear models (LQR) - fundamental result

under the simplifying assumption that the systems that we consider are fully controllable

Theorem

If

$$\begin{cases} \dot{\boldsymbol{x}} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} \end{cases} \qquad J(y, u) = \rho \|y\|_2^2 + \|u\|_2^2$$

then

$$\arg\min_{u \in \mathbb{R}_u} J(y, u) = -K\boldsymbol{x}$$

for an opportune K.

How can we find K? Follow the classical algorithms

Control with linear models – from LQR to MPC

What if:

$$\arg\min_{u\in\mathcal{U},y\in\mathcal{Y}} J(y,u) = \rho\,\|y\|_2^2 + \|u\|_2^2 \quad \text{s.t.} \begin{cases} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + Bu \\ y &= C\boldsymbol{x} \end{cases} ?$$

Control with nonlinear models (NL-MPC)

$$
\begin{aligned}
&\arg\min_{u} \quad && J\left(y, u\right) \\
&\text{s.t.} && \begin{cases} \dot{\boldsymbol{x}} &= f\left(\boldsymbol{x}, u, \theta\right) \\ y &= g\left(\boldsymbol{x}, u, \theta\right) \end{cases} \\
& && u \in \mathcal{U} \\
& && y \in \mathcal{Y}
\end{aligned}
$$

Main messages up to now

we need a control-oriented

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= f\left(\boldsymbol{x}, u, \theta\right) \\
y &= g\left(\boldsymbol{x}, u, \theta\right)
\end{aligned}
$$

and we need to have a good guess for $f(\cdot)$, $g(\cdot)$, and θ

how do we create a control-oriented model?

Yet an other way of categorizing models

white box: get structure from physics, get parameters from datasheets

grey box: get structure from physics, get parameters using system identification

black box: get both structure and parameters using system identification

The simplest non-white model: ARX

$$y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \ldots + b_{n_b} u(t-n_b) + e(t)$$

$$\theta = \left[a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}\right]^T \qquad e(t) \sim \mathcal{N}\left(0, \sigma^2\right)$$

The simplest non-white model: ARX

$$y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \ldots + b_{n_b} u(t-n_b) + e(t)$$

$$\theta = [a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}]^T \qquad e(t) \sim \mathcal{N}\left(0, \sigma^2\right)$$

Notation:

- $A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}$
- $B(q) = b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b}$

$$\Longrightarrow \qquad A(q)y(t) = B(q)u(t) + e(t)$$

Why "ARX"?

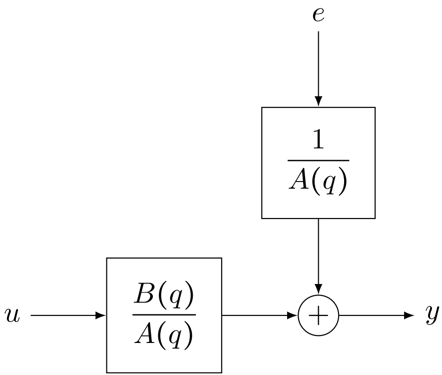
$$A(q)y(t) = B(q)u(t) + e(t)$$

AR: $A(q)y(t)$ *(autoregressive)*

X: $B(q)u(t)$ *(exogenous)*

ARX models - block schematic representation

$$A(q)y(t) = B(q)u(t) + e(t) \quad \Longrightarrow \quad y(t) = \frac{B(q)}{A(q)}u(t) + \frac{1}{A(q)}e(t) \quad \Longrightarrow$$



Towards more complex models: ARMAX

$$y(t)+a_1y(t-1)+\ldots+a_{n_a}y(t-n_a) = b_1u(t-1)+\ldots+b_{n_b}u(t-n_b)+e(t)+c_1e(t-1)+\ldots+c_{n_c}e(t-n_c)$$

$$\theta = \left[a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}, c_1, \ldots, c_{n_c}\right]^T \qquad e(t) \sim \mathcal{N}\left(0, \sigma^2\right)$$

Towards more complex models: ARMAX

$$y(t)+a_1y(t-1)+\ldots+a_{n_a}y(t-n_a) = b_1u(t-1)+\ldots+b_{n_b}u(t-n_b)+e(t)+c_1e(t-1)+\ldots+c_{n_c}e(t-n_c)$$

$$\theta = [a_1,\ldots,a_{n_a},b_1,\ldots,b_{n_b},c_1,\ldots,c_{n_c}]^T \qquad e(t)\sim\mathcal{N}\left(0,\sigma^2\right)$$

Notation:

- $A(q) = 1 + a_1q^{-1} + \ldots + a_{n_a}q^{-n_a}$
- $B(q) = b_1q^{-1} + \ldots + b_{n_b}q^{-n_b}$
- $C(q) = 1 + c_1q^{-1} + \ldots + c_{n_c}q^{-n_c}$

$$\Longrightarrow \qquad A(q)y(t) = B(q)u(t) + C(q)e(t)$$

Why “ARMAX”? (name)

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

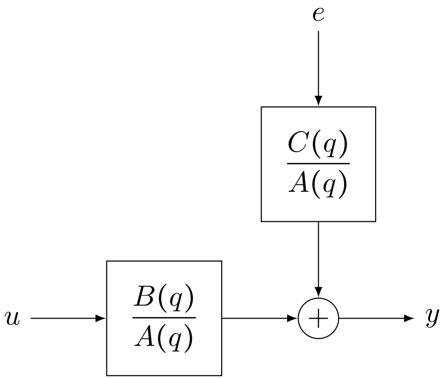
AR: $A(q)y(t)$ *(autoregressive)*

MA: $C(q)e(t)$ *(moving-average)*

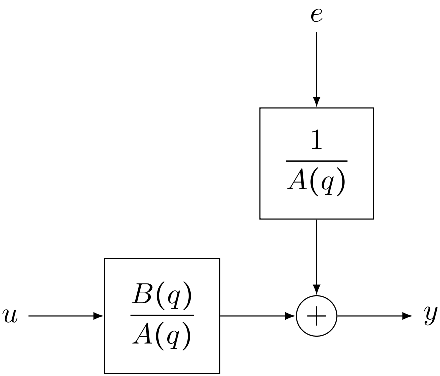
X: $B(q)u(t)$ *(exogenous)*

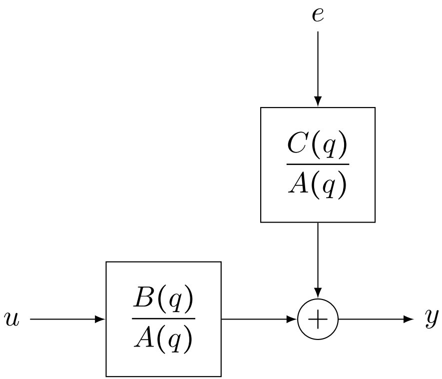
ARMAX models - block schematic representation

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad \Longrightarrow \quad y(t) = \frac{B(q)}{A(q)}u(t) + \frac{C(q)}{A(q)}e(t) \quad \Longrightarrow$$



Limitations of ARX and ARMAX models

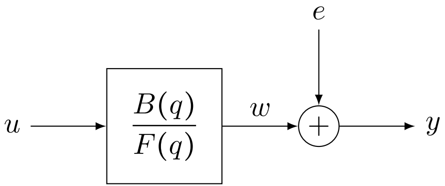




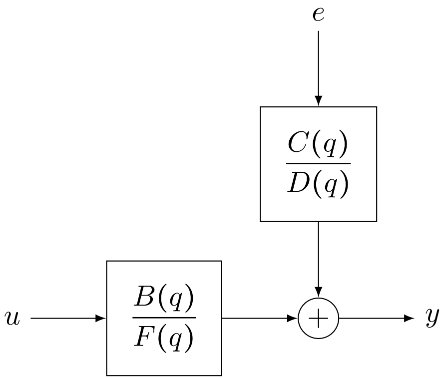
$A(q)$ = *denominator for both transfer functions*

(kind of limiting)

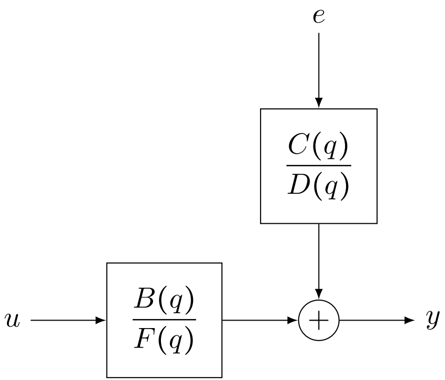
Output Error (OE) = simplest digression from ARX and ARMAX



Box-Jenkins (BJ) $=$ more sophisticated digression from ARX and ARMAX



Box-Jenkins (BJ) = more sophisticated digression from ARX and ARMAX



very general, often impractical

(more general models $\Longrightarrow$ more difficult estimation process)

So: how do we actually create a control-oriented model?

Typical strategy:

1. collect data
2. try to identify a linear model (ARX, ARMAX, ...)
3. see if it has good predictive capabilities
4. if so, do a linear controller
5. if not, try nonlinear identification and nonlinear control

how do we identify a system from the data?

(linear or nonlinear, in the next few slides it doesn't matter)

Preliminary step: Least-Squares

i.e., the simplest strategy for estimating parameters from collected data

Assumptions:

data generation model: $y_t = f(u_t; \theta) + v_t$

dataset: $\mathcal{D} = \{(u_t, y_t)\}_{t=1,\ldots,N}$

hypothesis space: $\theta \in \Theta$

Preliminary step: Least-Squares

i.e., the simplest strategy for estimating parameters from collected data

Assumptions:

data generation model: $y_t = f(u_t; \theta) + v_t$

dataset: $\mathcal{D} = \{(u_t, y_t)\}_{t=1,\ldots,N}$

hypothesis space: $\theta \in \Theta$

Problem: *find* θ *that "best explains"* $\mathcal{D}$

Least-squares: geometrical interpretation

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \qquad \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \qquad \begin{bmatrix} f\left(u_1; \theta\right) \\ \vdots \\ f\left(u_N; \theta\right) \end{bmatrix}$$

Least-squares: mathematical formulation

$$y_t = f\left(u_t; \theta\right) + v_t \qquad \mathcal{D} = \left\{\left(u_t, y_t\right)\right\}_{t=1,\ldots,N}$$

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} f\left(u_1; \theta\right) \\ \vdots \\ f\left(u_N; \theta\right) \end{bmatrix} \right\|^2$$

Least-squares: mathematical formulation

$$y_t = f\left(u_t; \theta\right) + v_t \qquad \mathcal{D} = \left\{\left(u_t, y_t\right)\right\}_{t=1,\dots,N}$$

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} f\left(u_1; \theta\right) \\ \vdots \\ f\left(u_N; \theta\right) \end{bmatrix} \right\|^2 = \arg\min_{\theta \in \Theta} \sum_{t=1}^{N} \Big(y_t - f\left(u_t; \theta\right) \Big)^2$$

Least-squares example: regression line

$$y_t = \theta_1 + \theta_2 u_t + v_t \qquad \mathcal{D} = \{(u_t, y_t)_t\} = \{(1,1),(2,2),(3,1)\} \qquad \theta \in \mathbb{R}^2$$

$$\widehat{\theta} = \arg\min_{\theta_1,\theta_2 \in \mathbb{R}} \Big((1-\theta_1-\theta_2)^2 + (2-\theta_1-2\theta_2)^2 + (1-\theta_1-3\theta_2)^2 \Big)$$

Main messages from the last few slides

- ARX, ARMAX, OE, BJ are simple control-oriented models
- doing system identification means estimating their parameters
- "estimation" actually means "optimization"

Main messages from the last few slides

- ARX, ARMAX, OE, BJ are simple control-oriented models
- doing system identification means estimating their parameters
- "estimation" actually means "optimization"

how do we identify ARX, ARMAX, OE, BJ, and all the rest?

parametric estimation as a predictors tuning problem

Parameter estimation methods

Assumption:

$$\mathcal{M} = \text{ selected model structure, e.g., } \begin{cases} \text{ARX} \\ \text{ARMAX} \\ \text{OE} \\ \dots \end{cases}$$

Parameter estimation methods

Assumption:

$$\mathcal{M} = \text{ selected model structure, e.g., } \begin{cases} \text{ARX} \\ \text{ARMAX} \\ \text{OE} \\ \ldots \end{cases}$$

main idea: a control-oriented model is as good as it can predict observed data

Parameter estimation methods

Assumption:

$$\mathcal{M} = \text{ selected model structure, e.g., } \begin{cases} \text{ARX} \\ \text{ARMAX} \\ \text{OE} \\ \dots \end{cases}$$

main idea: a control-oriented model is as good as it can predict observed data

In the linear case:

$$y(t) = G(q;\theta)u(t) + H(q;\theta)e(t)$$

$$\Downarrow$$

$$\widehat{y}(t|t-1\ ;\ \theta) = \left[H^{-1}(q;\theta)G(q;\theta)\right]u(t) + \left[1 - H^{-1}(q;\theta)\right]y(t)$$

Parameter estimation methods

Assumption:

$$\mathcal{M} = \text{ selected model structure, e.g., } \begin{cases} \text{ARX} \\ \text{ARMAX} \\ \text{OE} \\ \dots \end{cases}$$

main idea: a control-oriented model is as good as it can predict observed data

In the linear case:

$$y(t) = G(q;\theta)u(t) + H(q;\theta)e(t)$$

$$\Downarrow$$

$$\widehat{y}(t|t-1\;;\;\theta) = \left[H^{-1}(q;\theta)G(q;\theta)\right]u(t) + \left[1 - H^{-1}(q;\theta)\right]y(t)$$

$$\Longrightarrow \quad \text{in general, best } \theta^* = \text{ that } \theta \text{ that "minimizes" } y(t) - \widehat{y}(t|t-1\;;\;\theta)$$

Prediction error methods in a nutshell

(and with some simplifications)

prediction errors: $\varepsilon(t;\theta) = y(t) - \widehat{y}(t|t-1\ ;\ \theta)$

Prediction error methods in a nutshell

(and with some simplifications)

prediction errors: $\varepsilon(t;\theta) = y(t) - \widehat{y}(t|t-1\ ;\ \theta)$

loss function: $V(\theta, \mathcal{D}) = \dfrac{1}{N}\sum_{t=1}^{N} \ell\left(\varepsilon_F(t;\theta)\right)$

Prediction error methods in a nutshell

(and with some simplifications)

prediction errors: $\varepsilon(t;\theta) = y(t) - \widehat{y}(t|t-1\ ;\ \theta)$

loss function: $V(\theta, \mathcal{D}) = \dfrac{1}{N}\sum_{t=1}^{N} \ell\left(\varepsilon_F(t;\theta)\right)$

PEM: $\widehat{\theta} = \arg\min_{\theta\in\Theta} V(\theta, \mathcal{D})$

PEM vs. machine learning

Special focus of PEM =

- minimize prediction errors
- consider dynamics and effects of feedback loops

PEM in practice through examples: identifying ARMAX models

$$A(q;\theta)y(t) = B(q;\theta)u(t) + C(q;\theta)e(t)$$

PEM in practice through examples: identifying ARMAX models

$$A(q;\theta)y(t) = B(q;\theta)u(t) + C(q;\theta)e(t)$$

$$\Downarrow$$

$$\varepsilon(t) = \frac{A(q;\theta)}{C(q;\theta)}y(t) - \frac{B(q;\theta)}{C(q;\theta)}u(t)$$

PEM in practice through examples: identifying ARMAX models

$$A(q;\theta)y(t) = B(q;\theta)u(t) + C(q;\theta)e(t)$$

$$\Downarrow$$

$$\varepsilon(t) = \frac{A(q;\theta)}{C(q;\theta)}y(t) - \frac{B(q;\theta)}{C(q;\theta)}u(t)$$

$$\Downarrow$$

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} \sum_{t=1}^{N} \ell\left(\frac{A(q;\theta)}{C(q;\theta)}y(t) - \frac{B(q;\theta)}{C(q;\theta)}u(t)\right)$$

how shall we implement it numerically?

PEM in practice through examples: identifying ARMAX models

through opportune rewritings:

$$\underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ c_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & c_n & \cdots & 1 \end{bmatrix}}_{=:\underline{C}} \underbrace{\begin{bmatrix} \varepsilon(1;\theta) \\ \vdots \\ \varepsilon(N;\theta) \end{bmatrix}}_{=:\boldsymbol{\varepsilon}} = \underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ a_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & a_n & \cdots & 1 \end{bmatrix}}_{=:\underline{A}} \underbrace{\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}}_{=:\boldsymbol{y}} - \underbrace{\begin{bmatrix} 0 & & & & \\ \vdots & \ddots & & & \\ b_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & b_n & \cdots & 0 \end{bmatrix}}_{=:\underline{B}} \underbrace{\begin{bmatrix} u(1) \\ \vdots \\ u(N) \end{bmatrix}}_{=:\boldsymbol{u}}$$

PEM in practice through examples: identifying ARMAX models

through opportune rewritings:

$$\underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ c_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & c_n & \cdots & 1 \end{bmatrix}}_{=:\underline{C}} \underbrace{\begin{bmatrix} \varepsilon(1;\theta) \\ \vdots \\ \varepsilon(N;\theta) \end{bmatrix}}_{=:\boldsymbol{\varepsilon}} = \underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ a_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & a_n & \cdots & 1 \end{bmatrix}}_{=:\underline{A}} \underbrace{\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}}_{=:\boldsymbol{y}} - \underbrace{\begin{bmatrix} 0 & & & & \\ \vdots & \ddots & & & \\ b_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & b_n & \cdots & 0 \end{bmatrix}}_{=:\underline{B}} \underbrace{\begin{bmatrix} u(1) \\ \vdots \\ u(N) \end{bmatrix}}_{=:\boldsymbol{u}}$$

$$\Longrightarrow \qquad \boldsymbol{\varepsilon} = \underline{C}^{-1}\underline{A}\boldsymbol{y} - \underline{C}^{-1}\underline{B}\boldsymbol{u}$$

PEM in practice through examples: identifying ARMAX models

through opportune rewritings:

$$\underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ c_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & c_n & \cdots & 1 \end{bmatrix}}_{=:\underline{C}} \underbrace{\begin{bmatrix} \varepsilon(1;\theta) \\ \vdots \\ \varepsilon(N;\theta) \end{bmatrix}}_{=:\boldsymbol{\varepsilon}} = \underbrace{\begin{bmatrix} 1 & & & & \\ \vdots & \ddots & & & \\ a_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & a_n & \cdots & 1 \end{bmatrix}}_{=:\underline{A}} \underbrace{\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}}_{=:\boldsymbol{y}} - \underbrace{\begin{bmatrix} 0 & & & & \\ \vdots & \ddots & & & \\ b_n & & \ddots & & \\ & \ddots & & \ddots & \\ & & b_n & \cdots & 0 \end{bmatrix}}_{=:\underline{B}} \underbrace{\begin{bmatrix} u(1) \\ \vdots \\ u(N) \end{bmatrix}}_{=:\boldsymbol{u}}$$

$$\implies \quad \boldsymbol{\varepsilon} = \underline{C}^{-1}\underline{A}\boldsymbol{y} - \underline{C}^{-1}\underline{B}\boldsymbol{u}$$

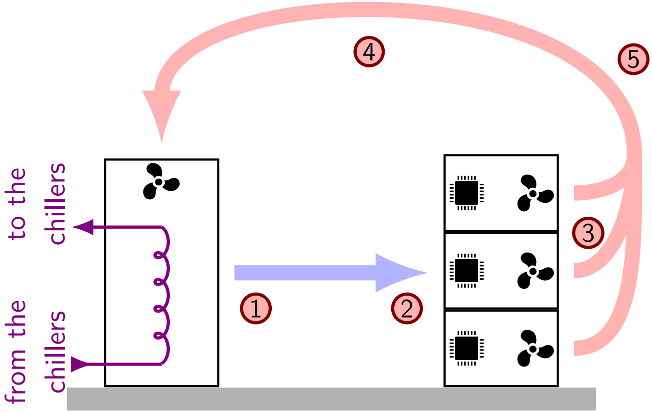
$$\implies \quad \arg \min_{\substack{a_1, \ldots, a_{n_a} \\ b_1, \ldots, b_{n_b} \\ c_1, \ldots, c_{n_c}}} V\left(\underline{C}^{-1}\underline{A}\boldsymbol{y} - \underline{C}^{-1}\underline{B}\boldsymbol{u}\right)$$

Main message from the last few slides

- identifying different model structures $\Longrightarrow$ implementing different optimization schemes

a practical example:
modelling air flow overprovisioning / underprovisioning

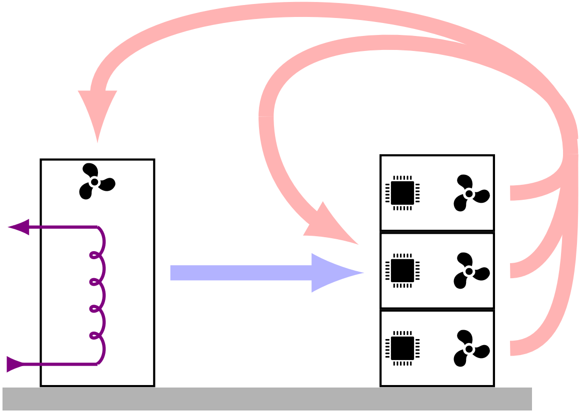
The ideal air flows distribution



= airflow temperature sensors

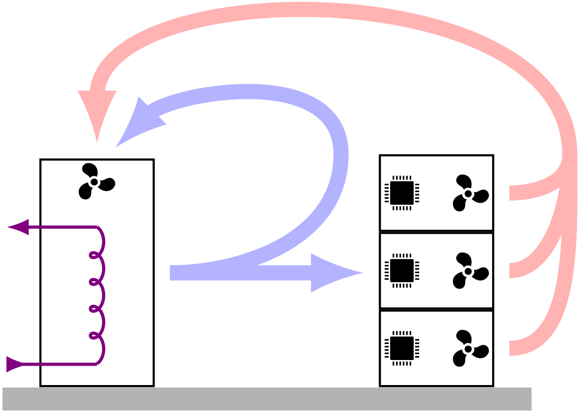
(notation: ideal provisioning $= \Omega_i$*)*

What do we mean with underprovisioning?



(notation: underprovisioning $= \Omega_u$*)*

What do we mean with overprovisioning?



(notation: overprovisioning $= \Omega_o$*)*

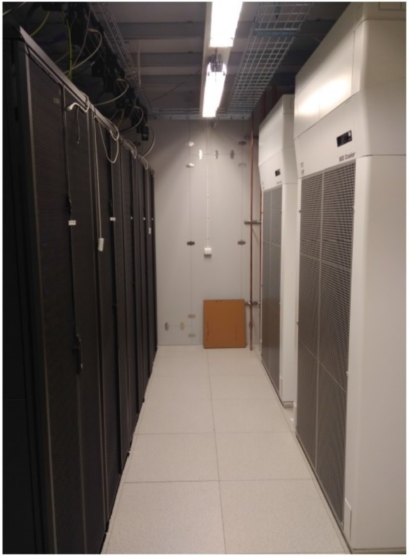
Generalizations

ideal provisioning := ventilation system working as planned

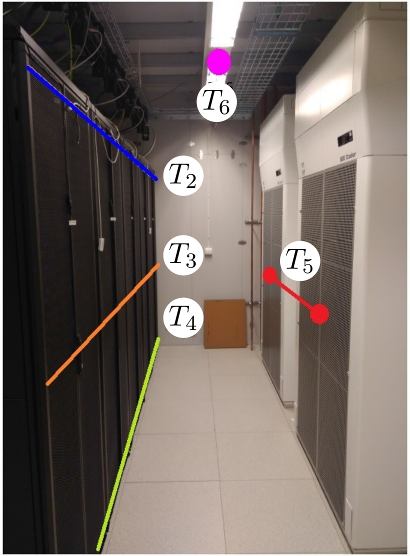
underprovisioning := servers receive warmer-than-ideal coolants

overprovisioning := cooling systems receive colder-than-ideal air intakes

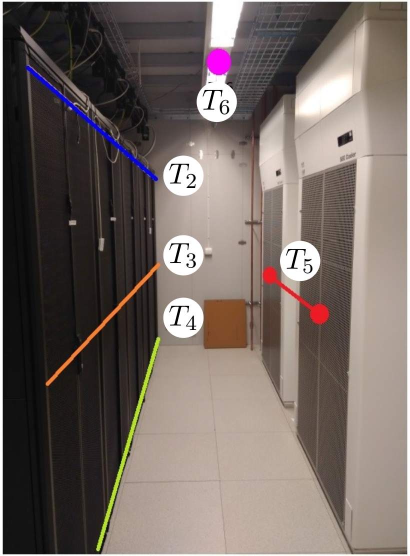
Towards data-driven modelling: which evidence is available?

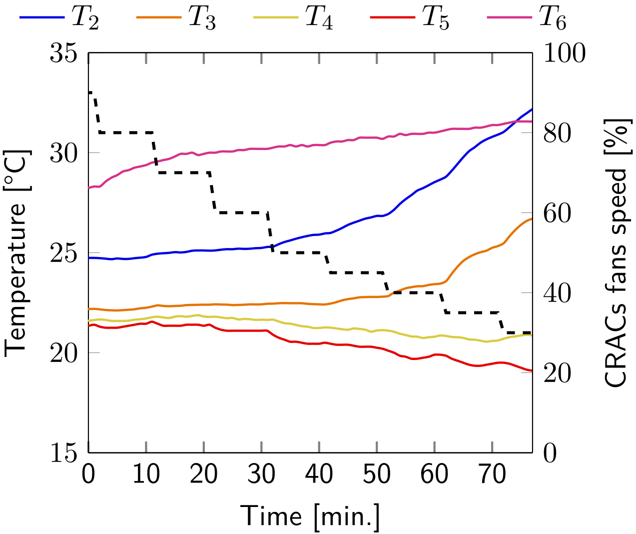


Towards data-driven modelling: which evidence is available?

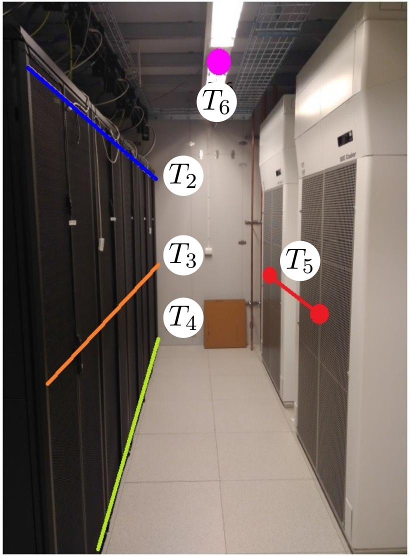


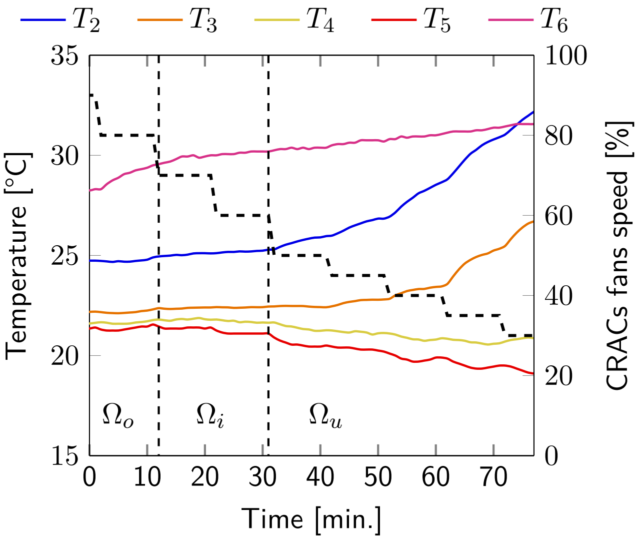
Towards data-driven modelling: which evidence is available?





Towards data-driven modelling: which evidence is available?





From developing the intuitions to modelling the system

From developing the intuitions to modelling the system

Choices:

- 3 Linear Time Invariant (LTI) models *(one for each provisioning region)*
- 2 alternative choices for the models configuration:
  - Single Input Single Output (SISO)
  - Multi Input Single Output (MISO)

Choice of the inputs and outputs

- SISO: $\begin{cases} \text{input:} & \text{CRACs fans speed} \\ \text{output:} & T_2 \text{ \textit{(topmost servers' air inlets temperature)}} \end{cases}$

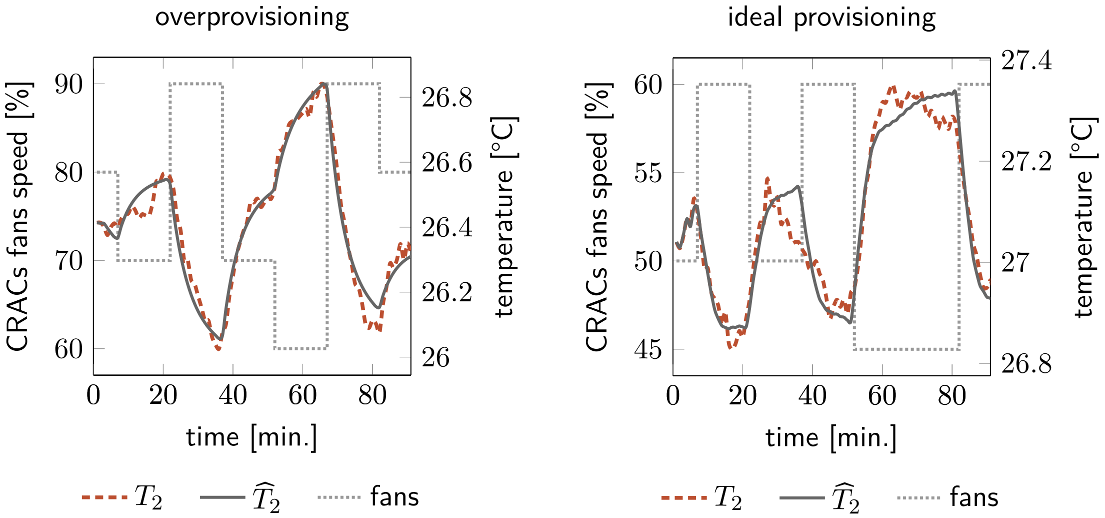
Choice of the inputs and outputs

- SISO: $\begin{cases} \text{input:} & \text{CRACs fans speed} \\ \text{output:} & T_2 \textit{ (topmost servers' air inlets temperature)} \end{cases}$

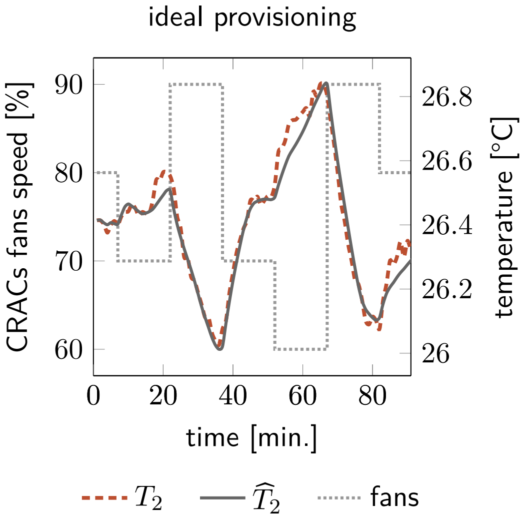
- MISO: $\begin{cases} \text{inputs:} & \text{CRACs fans speed, } T_6,\ T_r \\ \text{output:} & T_2 \textit{ (topmost servers' air inlets temperature)} \end{cases}$

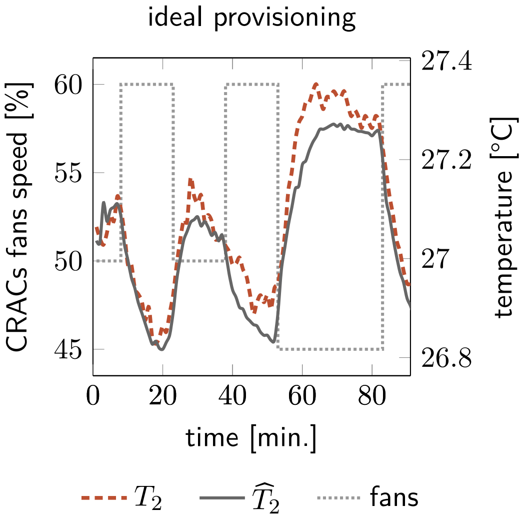
  with
  - T_6 = air temperature on the roof
  - $T_r = \dfrac{T_{\text{in}} + T_{\text{out}}}{2}$
  - T_{in} = temperature of the CRAC inlet refrigerant
  - T_{out} = temperature of the CRAC outlet refrigerant

Results - capabilities of the SISO model to simulate a validation dataset



Results - capabilities of the MISO model to simulate a validation dataset





Quantitative results

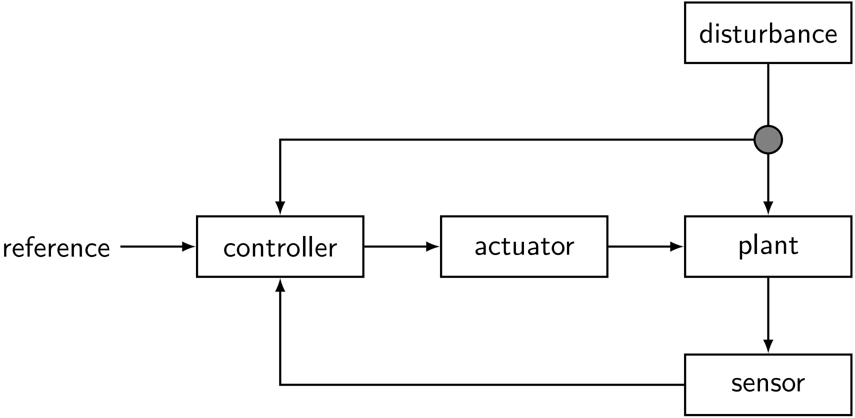
Provisioning region	Model type	Type	Order	Fit
over	SISO	BJ	$[3\,3\,2\,2]$	$81\,\%$
	MISO	ARX	$[2\,2\,2\,2]$	$83\,\%$
ideal	SISO	BJ	$[2\,2\,5\,5]$	$75\,\%$
	MISO	ARX	$[3\,3\,3\,3]$	$69\,\%$
under	SISO	BJ	$[2\,2\,5\,5]$	$85\,\%$
	MISO	ARX	$[3\,3\,3\,3]$	$88\,\%$

Ok, we got some models. So?

Ok, we got some models. So?

$\Longrightarrow$ possibilities for better airflow control

Conclusions



Control-oriented modelling - what is it?

Damiano Varagnolo

October 2018