Fast measurements of slow processes

Ivan Markovsky

University of Southampton

Setup

to-be-measured variable u $\xrightarrow{\text{measurement process}}$ measured variable y

- the measurement process is a dynamical system
- assumption 1: measured variable is a constant $u(t) = \bar{u}$
  (can be relaxed to "u's change is slower than y's change")
- y is a function of time and depends on both
  - measurement device dynamics and
  - environment dynamics
- assumption 2: measurement process is stable LTI system

Example 1: temperature measurement

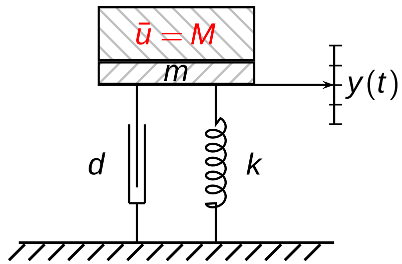
environmental temperature $\bar{u}$ $\xrightarrow{\text{environment–thermometer heat transfer}}$ thermometer's reading y

- measurement process: Newton's law of cooling

$$\frac{\mathrm{d}}{\mathrm{d}t} y = a(\bar{u} - y)$$

- the heat transfer coefficient $a > 0$
  depends on thermometer and environment
- first order stable LTI system
- dc-gain of measurement process is 1 (independent of a)

Example 2: weight measurement



- measurement process

$$(M + m)\frac{\mathrm{d}^2}{\mathrm{d}t^2} y + d\frac{\mathrm{d}}{\mathrm{d}t} y + ky = g\bar{u}$$

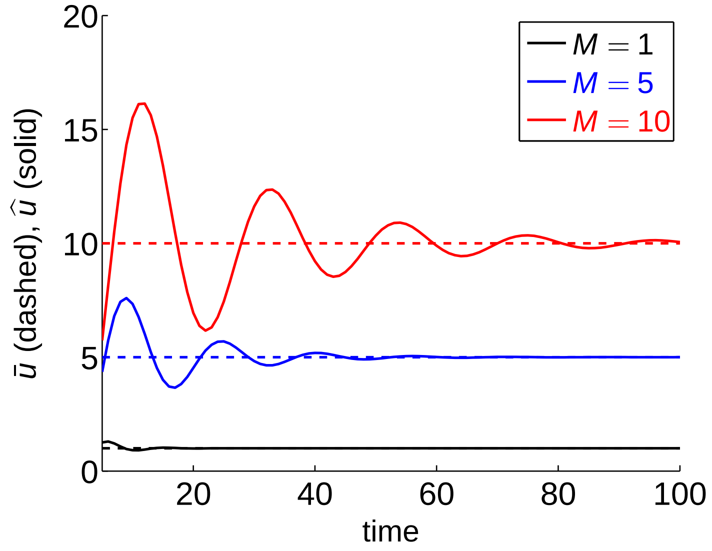
- the measurement process dynamics depends on M
- the dc-gain is $-g/k$ (independent of M)

Naive measurement

- assumption 3: measurement process's dc-gain G is known and nonzero (full column rank in the multivariable case)
- ignore the dynamics; consider the process as static system

$$\widehat{u}(t) := G^{-1}y(t)$$

- by the stability assumption, $\widehat{u}(t) \to \bar{u}$ as $t \to \infty$
- in reality, one waits for the transient to die out before reading the sensor measurement
- how much one needs to wait depends on the process



Dynamic measurement: basic idea

- process the data y in real-time aiming to predict $\bar{u}$
- problem: find system F, such that $\widehat{u} := Fy \approx \bar{u}$
- let H be process dynamics' transfer function; with $F = H^{-1}$

$$\widehat{u} = Fy = H^{-1}y = H^{-1}H\bar{u} = \bar{u}$$

Dynamic measurement: basic idea

- process the data y in real-time aiming to predict $\bar{u}$
- problem: find system F, such that $\widehat{u} := Fy \approx \bar{u}$
- let H be process dynamics' transfer function; with $F = H^{-1}$

$$\widehat{u} = Fy = H^{-1}y = H^{-1}H\bar{u} = \bar{u}$$

- F has to be causal

Dynamic measurement: basic idea

- process the data y in real-time aiming to predict $\bar{u}$
- problem: find system F, such that $\widehat{u} := Fy \approx \bar{u}$
- let H be process dynamics' transfer function; with $F = H^{-1}$

$$\widehat{u} = Fy = H^{-1}y = H^{-1}H\bar{u} = \bar{u}$$

- F has to be causal, perform "well" in presence of noise

Dynamic measurement: basic idea

- process the data y in real-time aiming to predict $\bar{u}$
- problem: find system F, such that $\widehat{u} := Fy \approx \bar{u}$
- let H be process dynamics' transfer function; with $F = H^{-1}$

$$\widehat{u} = Fy = H^{-1}y = H^{-1}H\bar{u} = \bar{u}$$

- F has to be causal, perform "well" in presence of noise, we care about transient due to nonzero initial conditions
- dynamic measurement with known process dynamics:
  1. off-line: design causal compensator F
  2. on-line: filter the data with F

Dynamic measurement: state-of-the-art

- with unknown measurement process dynamics, the approach being used in the literature is to on-line:
  - identify the process dynamics
  - tune the filter F according to the process parameters
  - filter the data with F
- computational requirements become an issue for implementation on DSP or specialised circuits
- as a result the developed solutions are specialised for particular application

Goals/results of this research

- generic solution for high order multivariable processes
  $\rightsquigarrow$ application of linear algebra and system theory
- address the problem as an input estimation problem without a priori bias towards a particular type of solution
  $\rightsquigarrow$ data-driven estimation algorithm
  (no need of on-line identification and filter tuning)
- treat noisy measurements in a statistically optimal way
  $\rightsquigarrow$ Kalman filter in case of known process dynamics, structured total least-squares otherwise

Problem formulation

given output observations

$$y = \big(y(t_1),\dots,y(t_T)\big), \qquad y(t) \in \mathbb{R}^p$$

of stable LTI system with dc-gain $G \in \mathbb{R}^{p\times m}$ and step input

find the input step value $\bar{u} \in \mathbb{R}^m$

noisy observations model:

$$y = y_0 + \widetilde{y} \quad \text{where} \quad \begin{array}{l} y_0 \text{ is exact trajectory} \\ \widetilde{y} \text{ is zero mean white Gaussian} \\ \quad \text{measurement noise} \end{array} \qquad (*)$$

Reduction to state estimation

$(\bar{u}s, y)$ is an input/output trajectory of nth order LTI system

$$\Updownarrow$$

y is a trajectory of autonomous $(n+m)$th order LTI system with m poles at 0 (continuous-time) or at 1 (discrete-time)

let $(\sigma x)(t) := x(t+1)$ and, in the discrete-time case, let

$$\mathscr{B} = \mathscr{B}_{\mathrm{ss}}(A,B,C,D) := \{\, w = (u,y) \mid \exists x,\ \sigma x = Ax + Bu \\ y = Cx + Du \,\}$$

be the I/O system; the corresponding autonomous system is

$$\mathscr{B}_{\mathrm{aut}} = \mathscr{B}_{\mathrm{ss}}(A_{\mathrm{aut}}, C_{\mathrm{aut}}) := \left\{ y \mid \exists x,\ \sigma x_{\mathrm{aut}} = \begin{bmatrix} A & B \\ 0 & I_m \end{bmatrix} x_{\mathrm{aut}},\ y = \begin{bmatrix} C & D \end{bmatrix} x_{\mathrm{aut}} \right\}$$

Proof

$$(\bar{u}s, y) \in \mathscr{B} = \mathscr{B}_{\mathrm{ss}}(A,B,C,D)$$

$$\iff \quad \sigma x = Ax + B\bar{u}s,\ y = Cx + D\bar{u}s, \quad x(0) = x_{\mathrm{ini}}$$

$$\iff \quad \sigma x = Ax + B\bar{u}s,\ \sigma\bar{u} = \bar{u},\ y = Cx + D\bar{u}s, \quad x(0) = x_{\mathrm{ini}}$$

$$\iff \quad \sigma x_{\mathrm{aut}} = A_{\mathrm{aut}} x_{\mathrm{aut}},\ y = C_{\mathrm{aut}} x_{\mathrm{aut}}, \quad x_{\mathrm{aut}}(0) = (x_{\mathrm{ini}}, \bar{u})$$

$$\iff \quad y \in \mathscr{B}_{\mathrm{aut}} = \mathscr{B}_{\mathrm{ss}}(A_{\mathrm{aut}}, B_{\mathrm{aut}})$$

Algorithm for input est. with known model

- given $\mathscr{B} = \mathscr{B}_{\mathrm{ss}}(A,B,C,D)$, define

$$\mathscr{B}_{\mathrm{aut}} = \mathscr{B}_{\mathrm{ss}}\left(\begin{bmatrix} A & B \\ 0 & I_m \end{bmatrix}, \begin{bmatrix} C & D \end{bmatrix} \right)$$

- (off-line) design a state estimator for $\mathscr{B}_{\mathrm{aut}}$
  - deadbeat observer (for exact data) or
  - Kalman filter (for noisy data)
- (on-line) process y with the state estimator $\rightsquigarrow \widehat{x}_{\mathrm{aut}} = \begin{bmatrix} \widehat{x} \\ \widehat{u} \end{bmatrix}$
- prior knowledge (mean and variance) about $x_{\mathrm{aut}}(0)$ can be used in the Kalman filtering algorithm

Comments

- deadbeat observer recovers $\bar{u}$ in at most $n+m$ samples
- Kalman filter is statistically optimal estimator in the case $(*)$
- the computational cost per sample is $O\big((n+m)^2\big)$ (assuming the Kalman filter gain is precomputed)
- no new theory; just application of existing one in new setup

The input est. problem with unknown model

given output observations

$$y = \big(y(t_1),\dots,y(t_T)\big), \qquad y(t) \in \mathbb{R}^p$$

of stable LTI system with dc-gain $G \in \mathbb{R}^{p\times m}$ and step input

find the input step value $\bar{u} \in \mathbb{R}^m$

resembles identification from step response data, except that

1. the input is unknown,
2. the dc-gain is constrained to be equal to G, and
3. the goal is to find $\bar{u}$ rather than the system dynamics

1 and 2 are easily dealt with, 3 leads to a data-driven solution

Reduction to step response estimation

$$(\bar{u}s, y) \text{ is trajectory of LTI system with dcgain } G \qquad (1)$$

$$\Updownarrow$$

$$\begin{array}{c}(\bar{u}'s, y) \text{ is trajectory of LTI system with dcgain } G' = PG \\ \text{where } P \text{ is } m\times m \text{ nonsingular matrix, such that } \bar{u} = P\bar{u}'\end{array} \qquad (2)$$

implication for input estimation: while in (1) $\bar{u}$ is unknown and G is given, in (2), we can choose $\bar{u}' \neq 0$ and treat G' as unknown

$\Longrightarrow$ input estimation problem with $p \geq m$ and unknown model is equivalent to identification from step response data $(\bar{u}'s, y)$

Algorithm based on identification from step response

Input: y and G

1. system identification: $(\mathbf{1}_m s, y) \mapsto \mathscr{B}'$, where $\mathbf{1}_m := \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} \in \mathbb{R}^m$
2. solve for $\bar{u}$ the system $G\bar{u} := \text{dcgain}(\mathscr{B}')\mathbf{1}_m$

Output: $\bar{u}$

- use output error identification in case of noisy data $(*)$
- optimal (maximum likelihood) identification $\Longrightarrow$ optimal estimation of $\bar{u}$
- recursive identification method $\Longrightarrow$ recursive method for estimation of $\bar{u}$

Reduction to autonomous system identification

$(s\bar{u}, y)$ is a trajectory of nth order LTI system with dcgain G

$$\Updownarrow$$

y is a trajectory of $(n+1)$st order autonomous system with pole at 0 (continuous-time) or 1 (discrete-time)

implication for input estimation: instead of modeling $(s\bar{u}, y)$ as response of nth order LTI system, one can model y as a response of $(n+1)$th order autonomous system with pole at 1

Proof

an output y of an LTI system $\mathscr{B}$ with input $u = \bar{u}s$ is of the form

$$y(t) = \left(\bar{y} + \sum_{i=1}^{n} \alpha_i \beta_i(t) z_i^t\right) s(t), \qquad \text{for all } t,$$

where $z_1, \ldots, z_n$ are $\mathscr{B}$'s poles, $\alpha_i \in \mathbb{R}^p$, and β_i are polynomials

it follows that y is a trajectory of an autonomous system

$$\mathscr{B}_{\text{ss}}\left(\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} C & d \end{bmatrix}\right)$$

How to ensure a pole at 1?

$$y \in \mathscr{B}_{\text{ss}}\left(\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} C & d \end{bmatrix}\right) =: \mathscr{B}_{\text{ss}}(A_{\text{e}}, C_{\text{e}})$$

$$\Updownarrow$$

$$\Delta y := (1 - \sigma^{-1}) y \in \Delta\mathscr{B} := \mathscr{B}_{\text{ss}}(A, C)$$
$$(\Delta y = y(t) - y(t-1))$$

Proof: let P be the characteristic polynomial of the matrix A

$$y \in \mathscr{B}_{\text{ss}}(A_{\text{e}}, C_{\text{e}}) \quad \Longleftrightarrow \quad P(\sigma^{-1})(1 - \sigma^{-1}) y = 0$$

on the other hand, we have

$$\Delta y := (1 - \sigma^{-1}) y \in \mathscr{B}_{\text{ss}}(A, C) \quad \Longleftrightarrow \quad P(\sigma^{-1})(1 - \sigma^{-1}) y = 0$$

How to find $\bar{u}$, given $\mathscr{B}_{\text{ss}}(A_{\text{e}}, C_{\text{e}})$?

once A and C are determined, $\bar{u}$ is computed from

$$y = \bar{y} + y_{\text{aut}}, \qquad \text{where} \quad \bar{y} = G\bar{u} \quad \text{and} \quad y_{\text{aut}} \in \mathscr{B}_{\text{ss}}(A, C)$$

or

$$\begin{bmatrix} G & C \\ G & CA \\ \vdots & \vdots \\ G & CA^{T-1} \end{bmatrix} \begin{bmatrix} \bar{u} \\ x_{\text{ini}} \end{bmatrix} = \begin{bmatrix} y(t_s) \\ \vdots \\ y(Tt_s) \end{bmatrix} \qquad (**)$$

Algorithm based on autonomous system identification

Input: y and G

1. compute the finite differences $\Delta y := (1-\sigma^{-1})y$
2. autonomous system identification: $\Delta y \mapsto \Delta\mathscr{B}$
3. compute $\bar{u}$ by solving $(**)$

Output: $\bar{u}$

- optimal (maximum likelihood) identification
  $\implies$ optimal estimation of $\bar{u}$
- recursive identification method
  $\implies$ recursive method for estimation of $\bar{u}$

Data-driven method

$$\Delta\mathscr{B} = \operatorname{span} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-n-1} \end{bmatrix}$$

$$= \operatorname{span} \underbrace{\begin{bmatrix} \Delta y(2) & \Delta y(3) & \cdots & \Delta y(n+1) \\ \Delta y(3) & \Delta y(4) & \cdots & \Delta y(n+2) \\ \Delta y(4) & \Delta y(5) & \cdots & \Delta y(n+3) \\ \vdots & \vdots & & \vdots \\ \Delta y(T-n) & \Delta y(T-n+1) & \cdots & \Delta y(T) \end{bmatrix}}_{\mathscr{H}_{T-n}(\Delta y)}$$

Data-driven algorithm

Input: y and G

1. compute the finite differences $\Delta y := (1-\sigma^{-1})y$
2. computed $\bar{u}$ by solving

$$\begin{bmatrix} \mathbf{1}_{T-n} \otimes G & \mathscr{H}_{T-n}(\Delta y) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \ell \end{bmatrix} = \begin{bmatrix} y\big((n+1)t_s\big) \\ \vdots \\ y\big(Tt_s\big) \end{bmatrix} \qquad (***)$$

Output: $\bar{u}$

- in the case of noisy data y, $(***)$ is solved approximately
- recursive least-squares method
  $\implies$ recursive method for estimation of $\bar{u}$
- $O\big((m+n)^2 p\big)$ computations per sample
  same order of magnitude as methods using given model

- with exact data, the estimate is exact, provided $T \geq 2n+m$ and G is full column rank
- the methods based on system identification require stronger (identifiability) considtion
- with noisy data, ML estimation requires approximate solution of $(***)$ in a structured total least-squares sense
- the (recursive) least-squares approximate solution yields a suboptimal estimate of $\bar{u}$

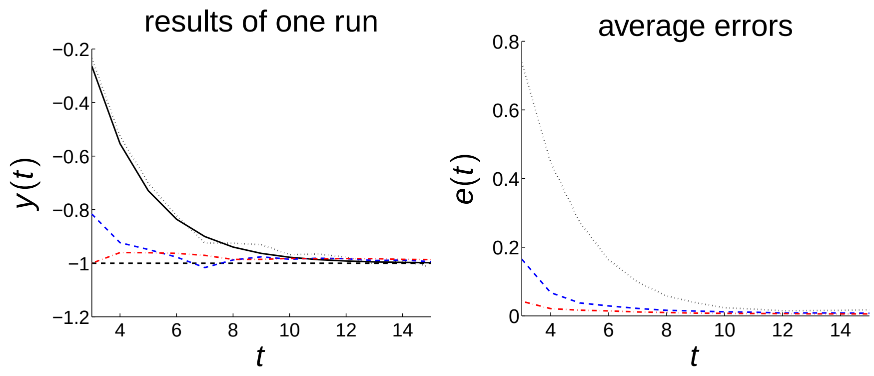
Testing

dashed	—	true parameter value $\bar{u}$
solid	—	true output trajectory y_0
dotted	—	naive estimate $\widehat{u} = G^{+}y$
dashed	—	Kalman filter
bashed-dotted	—	data-driven

estimation error: $e := \frac{1}{N}\sum_{i=1}^{N} \|\bar{u} - \widehat{u}^{(i)}\|_1 \qquad (\|x\|_1 := \sum_{i=1}^{n} |x_i|)$

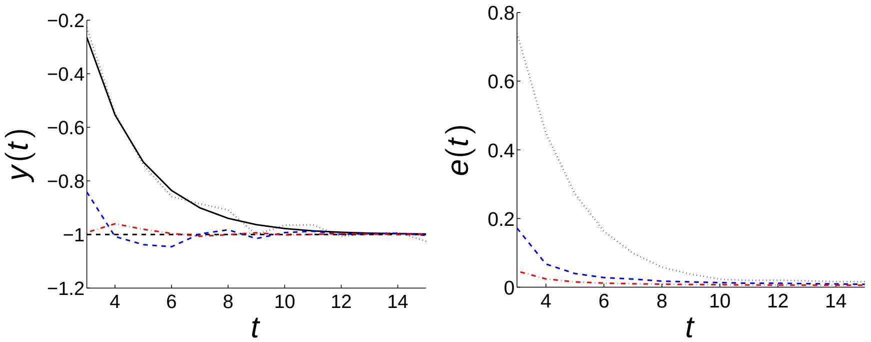
where $\widehat{u}^{(i)}(t)$ is an estimate of $\bar{u}$ using the data $y(1),\dots,y(t)$

Dynamic cooling $a = 0.5$, $x_{\text{ini}} = 1$, $\sigma = 0$



exact data $\Longrightarrow$ exact estimate after $2n + m = 3$ samples

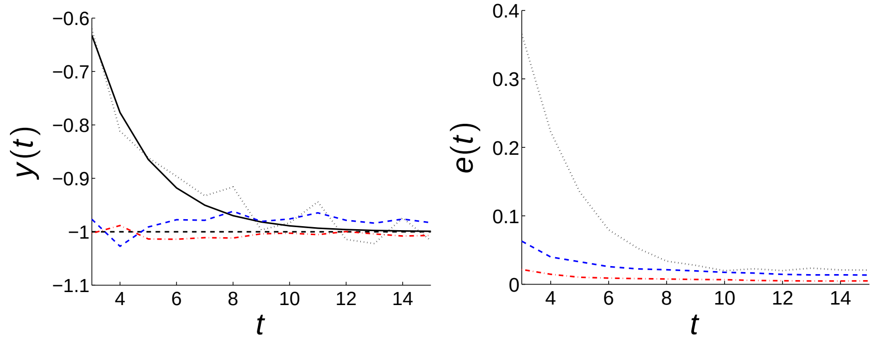
Dynamic cooling $a = 0.5$, $x_{\text{ini}} = 1$, $\sigma = 0.02$



noisy data $\Longrightarrow$ $e(t) \to 0$ as $t \to \infty$ (at different rates!)

note: Kalman filter is maximum likelihood estimator in this setup

Temperature and pressure sensors $\sigma_{\text{temp}} = 0.02$, $\sigma_{\text{pressure}} = 0.05$

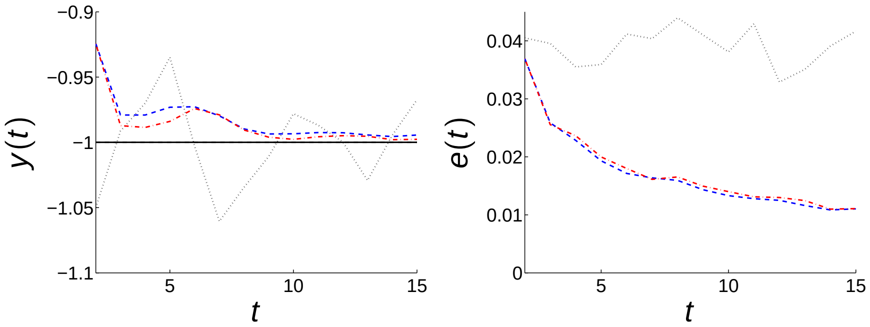


assuming constant volume and ideal gas

temperature $=$ constant $\times$ pressure

so properly calibrated pressure sensor measures temperature

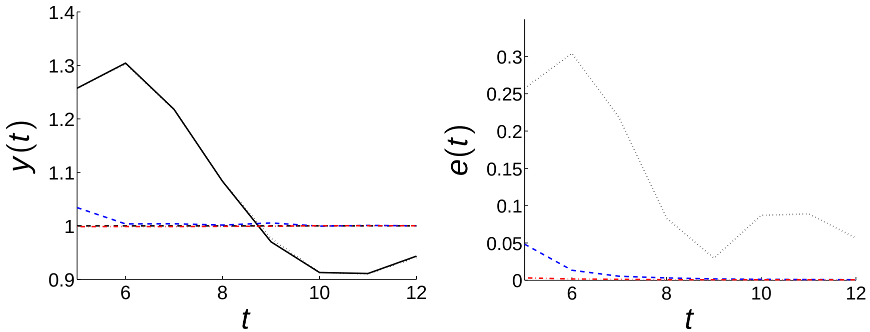
Pressure sensor only $\sigma = 0.05$



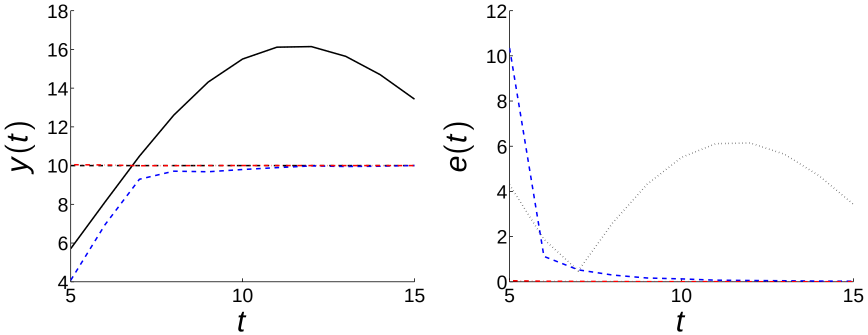
Note: in the noisy case, the methods give improvement in accuracy as well as speed

Dynamic weighing

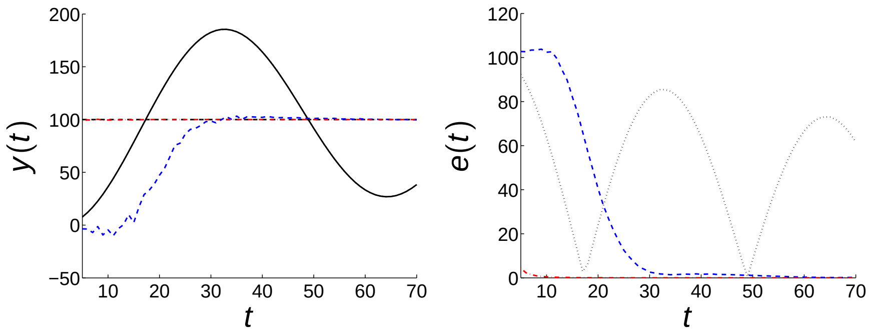
$m = 1$, $M = 1$, $k = 1$, $d = 1$, $x_{\text{ini}} = 0.1 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\sigma = 0.02$



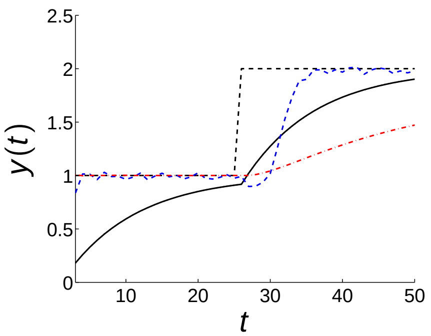
Dynamic weighing $M = 10$

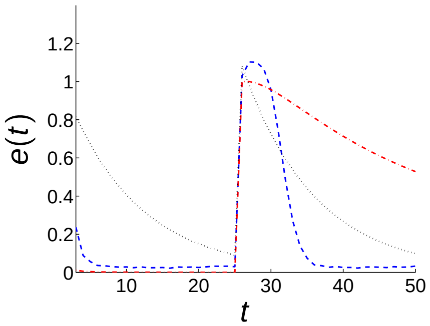


Dynamic weighing $M = 100$



Time-varying parameter





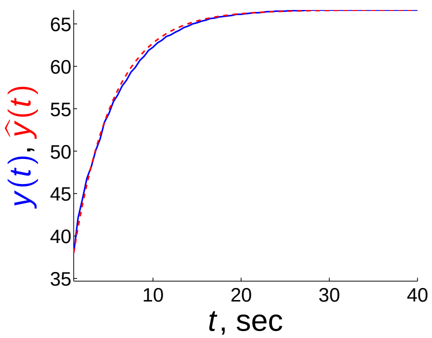
- dynamic cooling setup with a jump in the temperature $\bar{u}$
- exponentially weighted recursive least squares with forgetting factor $f = 0.5$

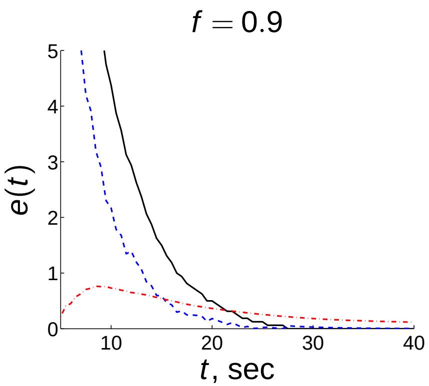
Experiment with Lego NXT Mindstorms

Results with real-life data

model for the KF is fitted using all measurements

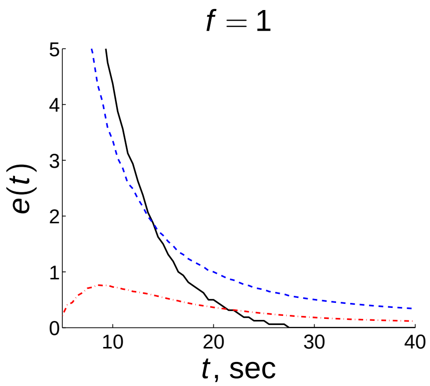
$t_s = 0.5$ sec, $\quad \bar{y} = \bar{u} := y(40)$

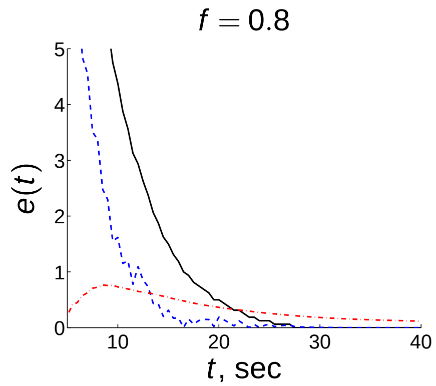




Results with real-life data

Q: Why $f = 0.9$? A: Gives better results (trail and error).





Conclusions

- methods for speeding up measurement devices
- improvement in both dynamical response and accuracy
- requirement: DSP attached to the sensor
- with a priori given model, optimal estimator is Kalman filter
- without model, standard identification methods are used
- main contribution: model-free algorithm, which is computationally as expensive as an LTI compensator
- link between step response and autonomous identification

Current/future work

- optimal data-driven algorithm (structured TLS problem)
- implementation and testing on DSP
- building laboratory prototypes (with Lego Mindstorms NXT)
- contact and get feedback from the metrology community
- contact and pursue uptake by industry

Questions?