

To model or not to model?

A tutorial on indirect data-driven control methods

Roy S. Smith

ETH Zürich
`rsmith@control.ee.ethz.ch`

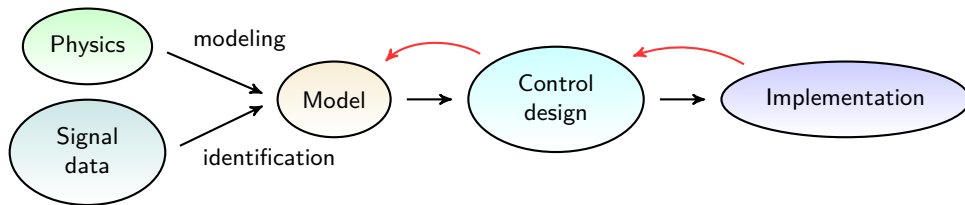
Collaborators: Mohamed Abdalmoaty, Mingzhou Yin, Andrea Iannelli.

14 October 2026

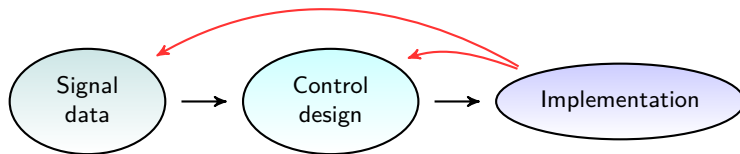
Low Level RF Workshop 2026
Nova Gorica, Slovenia

Control design and implementation pipelines

“Classical” approach

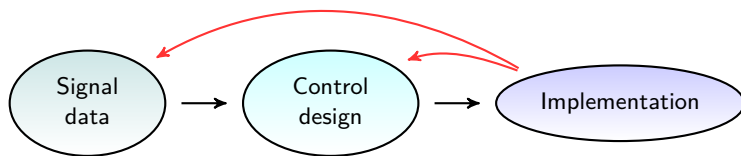


“Data-driven” approach



Motivations

Why might we be interested in using data-driven approaches?



The Push

- ▶ Model structure decisions?
- ▶ Selecting the identification method(s)?
- ▶ How to include the physics?
- ▶ Incompatible design and simulation domains?

The Pull

- ▶ Increasing availability of data
- ▶ Bypass modeling considerations (data determines model)
- ▶ Potential to adapt the design to new data
- ▶ Automate (semi-automate) design process

Outline

- ▶ Hankel matrix representations (Signal Matrix Models)
 - ▶ Model predictive control
 - ▶ Willems' lemma (noise-free model data)
 - ▶ Data enabled predictive control (DeePC) and variants
- ▶ Causal SMM formulation
 - ▶ Causal separation: initial condition estimation plus future prediction
 - ▶ Subspace predictive control
 - ▶ Illustrative example (747 pitch control) with noisy model data.
- ▶ Dynamic disturbance in the data
 - ▶ SMM-based predictors & Kalman filtering
 - ▶ Data-driven innovations-based predictors.
- ▶ Closing comments and recommendations

Model predictive control: optimisation-based implementation

Conceptual description: succesively planning the next control input sequence

1. Optimise over a length- T_f horizon into the future.

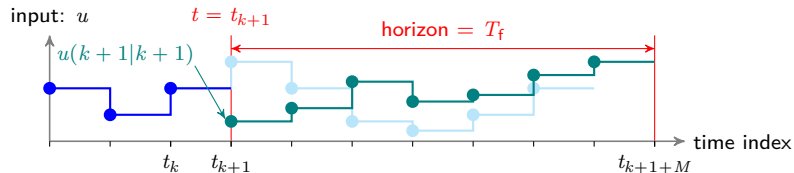
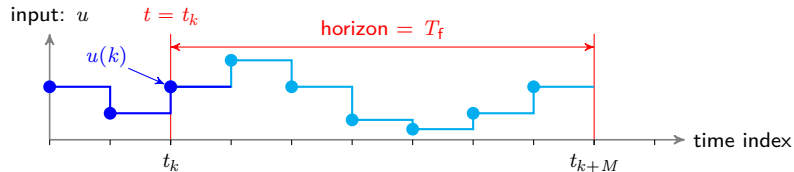
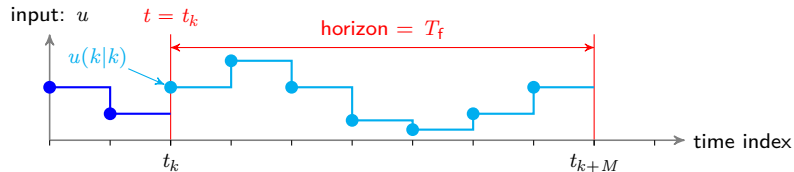
$$\underset{u, y}{\text{minimise}} \quad \sum_{k=0}^{T_f-1} J_{\text{cost}}(u, y) \quad \longleftarrow \text{Cost function over length-}T_f \text{ horizon}$$

subject to: u and y satisfying the *dynamic model of the system*

$$u \in \mathcal{U}, \quad y \in \mathcal{Y} \quad \longleftarrow \text{Operational constraints}$$

2. Implement the first step of the optimal u trajectory.
3. At the next time step collect new measurements.
4. Repeat step 1.

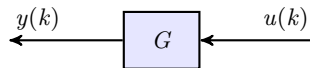
Model predictive control



Models: linear time-invariant systems (time-domain)

Pulse response

$$y(k) = \sum_{i=0}^{\infty} g(i)u(k-i)$$



Real-rational transfer functions

$$\begin{aligned} G &= \frac{B(z)}{A(z)} = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \dots b_m z^{n-m}}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots a_n} \\ &= \frac{b_1 z^{-1} + b_2 z^{-2} + \dots b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots a_n z^{-n}} \end{aligned}$$

State-space representations

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

Indirect “data-driven” formulation

Willems’ fundamental lemma (assumptions on dimension and rank)^{1 2}

Given a sufficiently exciting input-output data, $\{u(k), y(k)\}$, $k = 0, \dots, K - 1$,

Define,

$$H_u = \begin{bmatrix} u(1) & u(2) & \cdots & u(M) \\ u(2) & u(3) & \cdots & u(M+1) \\ \vdots & \vdots & & \vdots \\ u(T) & u(T+1) & \cdots & u(K-1) \end{bmatrix} \quad \text{and} \quad H_y = \begin{bmatrix} y(1) & y(2) & \cdots & y(M) \\ y(2) & y(3) & \cdots & y(M+1) \\ \vdots & \vdots & & \vdots \\ y(T) & y(T+1) & \cdots & y(K-1) \end{bmatrix}.$$

Then u, y is a length T input-output trajectory of the system **if and only if** there exists g such that,

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} H_u \\ H_y \end{bmatrix} g. \quad \text{Note: } \begin{bmatrix} H_u \\ H_y \end{bmatrix} \text{ has more columns than rows, and is low rank.}$$

¹Willems, Rapisarda, Markovsky, De Moor, SCL, 2005.

²De Persis & Tesi, TAC, 2019.

Prediction directly from data matrices (Signal Matrix Models)

Using data as a model³

Split signals and Hankel data matrices into past and future ($T = T_p + T_f$).

$$u = \left[\begin{array}{l} u_p \\ u_f \end{array} \right] \left\{ \begin{array}{l} \leftarrow \text{length } T_p \\ \leftarrow \text{length } T_f \end{array} \right\} \text{length } T. \quad y = \left[\begin{array}{l} y_p \\ y_f \end{array} \right], \quad H_u = \left[\begin{array}{l} H_{up} \\ H_{uf} \end{array} \right], \quad H_y = \left[\begin{array}{l} H_{yp} \\ H_{yf} \end{array} \right],$$

Output prediction, y_f

Given past data, u_p & y_p , and a future input u_f we can calculate y_f by finding any g solving,

$$\left[\begin{array}{l} u_p \\ u_f \\ y_p \\ y_f \end{array} \right] = \left[\begin{array}{l} H_{up} \\ H_{uf} \\ H_{yp} \\ H_{yf} \end{array} \right] g. \quad \text{One solution: } g = \left[\begin{array}{l} H_{up} \\ H_{uf} \\ H_{yp} \end{array} \right]^\dagger \left[\begin{array}{l} u_p \\ u_f \\ y_p \end{array} \right], \quad y_f = H_{yf} g.$$

³Markovsky & Rapisarda, IJC, 2008.

Data-driven control

Model predictive control

Control objective:

$$\underset{g, u_f, y_f}{\text{minimise}} \quad (y_f - r)^T Q (y_f - r) + u_f^T R u_f^T$$

$$\text{subject to: } \begin{bmatrix} u_p \\ u_f \\ y_p \\ y_f \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{uf} \\ H_{yp} \\ H_{yf} \end{bmatrix} g, \quad u_f \in \mathcal{U}, \quad y_f \in \mathcal{Y}.$$

← Matches initial conditions
Predicts y_f as a function of u_f

Data-driven control

Model predictive control (DeePC)⁴

Control objective:

$$\underset{g, u_f, y_f, \sigma_y}{\text{minimise}} \quad (y_f - r)^T Q (y_f - r) + u_f^T R u_f^T + \lambda_y \|\sigma_y\| + \lambda_g \|g\|$$

$$\text{subject to:} \quad \begin{bmatrix} u_p \\ u_f \\ y_p + \sigma_y \\ y_f \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{uf} \\ H_{yp} \\ H_{yf} \end{bmatrix} g, \quad u_f \in \mathcal{U}, \quad y_f \in \mathcal{Y}.$$

Similar data-driven methods:

γ -DDPC,⁵ Generalized data predictive control (GDPC).⁶

⁴Coulson, Lygeros, Dörfler, *ECC*, 2019.

⁵Breschi, Chiuso, Formentin, *Automatica*, 2023

⁶Lazar & Verheijen, *Mathematics*, 2023

Data-driven control

Signal data sources:

- ▶ High fidelity simulation (digital twin)? (Ideal)
- ▶ Dedicated experiment? (Noisy. Experiment design minimises problems)
- ▶ Operational or historical data? (Noisy. Persistency of excitation problems)

Noisy model data

Noisy data in H_y (or H_u) can be problematic.

$\begin{bmatrix} H_u \\ H_y \end{bmatrix}$ becomes full rank so $\begin{bmatrix} u_p \\ y_p \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{yp} \end{bmatrix} g$ does not constrain g .

- ▶ Regularisation becomes more important for DeePC/ γ -DDPC/GDPC⁷.
- ▶ Maximum likelihood estimation of g is an alternative⁸.

⁷Mattsson & Schön, IFAC WC, 2023

⁸Yin, Iannelli, Smith, IEEE TAC, 2022

Decoupling the initial condition estimation and the control

A causal decoupling

$$\begin{bmatrix} u_p \\ u_f \\ y_p \\ y_f \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{uf} \\ H_{yp} \\ H_{yf} \end{bmatrix} g \quad \longleftrightarrow \quad \begin{bmatrix} u_p \\ u_f \\ y_p \\ y_f \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{uf} \\ H_{yp} \\ H_{yf} \end{bmatrix} (g_p + g_f) \quad \text{with } g_p \perp g_f.$$

g_p gives the best match to the observed past data, u_p, y_p .

g_f matches a desired output, y_f , with the required input, u_f .

Key point: g_f does not affect u_p or y_p .

so now: g_p is chosen to match the initial conditions.

g_f is chosen to satisfy a control objective.

Predictive control

Exploiting the causal decoupling

Control problem:

$$\begin{aligned} & \underset{g_p, g_f, u_f, y_f}{\text{minimise}} && (y_f - r)^T Q (y_f - r) + u_f^T R u_f \\ & \text{subject to:} && \begin{bmatrix} u_p \\ y_p \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{yp} \end{bmatrix} g_p \\ & && \begin{bmatrix} u_f \\ y_f \end{bmatrix} = \begin{bmatrix} H_{uf} \\ H_{yf} \end{bmatrix} (g_p + g_f), \quad u_f \in \mathcal{U}, \quad y_f \in \mathcal{Y}. \end{aligned} \tag{1}$$

In the noisy measurement case we solve constraint (1) in a least-squares sense.

Matching the past data

A more parsimonious representation⁹

Consider the past signals.

$$H_{yup} = \begin{bmatrix} H_{up} \\ H_{yp} \end{bmatrix} \quad \text{and, via a QR decomposition,} \quad H_{yup} = \begin{bmatrix} L_{up} & 0 & 0 \\ L_{yp} & L_{yp} & 0 \end{bmatrix} \begin{bmatrix} Q_{up}^T \\ Q_{yp}^T \\ Q_{np}^T \end{bmatrix},$$

(L_{up} and L_{yp} are lower triangular)

Given u_p and y_p we can solve,

$$\begin{bmatrix} u_p \\ y_p \end{bmatrix} = \begin{bmatrix} H_{up} \\ H_{yp} \end{bmatrix} g_p = \begin{bmatrix} L_{up} & 0 \\ L_{yp} & L_{yp} \end{bmatrix} \begin{bmatrix} x_u \\ x_y \end{bmatrix}. \quad \text{Here } g_p = [Q_{up} \quad Q_{yp}] \begin{bmatrix} x_u \\ x_y \end{bmatrix}.$$

⁹Smith, Abdalmoaty, Yin, CDC, 2024a

Predicting the future data

Parametrise the future responses within the null-space of the past.

Using an LQ decomposition, $H_{yuf}Q_{np} = L_f Q_f^T = \left[\begin{array}{c|c} L_{uf} & 0 \\ \hline L_{yf} & 0 \end{array} \right] \left[\begin{array}{c} Q_{yuf}^T \\ Q_{nf}^T \end{array} \right],$

Given a future input u_f , the variable z solving

$$u_f = L_{uf} z \quad \text{uniquely determines the causal response,} \quad y_f = L_{yf} z.$$

A smaller decomposition for Willems' lemma

$$\left[\begin{array}{c} u_p \\ y_p \\ \hline u_f \\ y_f \end{array} \right] = \left[\begin{array}{cc|c} L_{up} & 0 & 0 \\ L_{yup} & L_{yp} & 0 \\ \hline S_{uu} & S_{uy} & L_{uf} \\ S_{yu} & S_{yy} & L_{yf} \end{array} \right] \left[\begin{array}{c} x_u \\ x_y \\ \hline z \end{array} \right], \quad \text{where} \quad \begin{bmatrix} S_{uu} & S_{uy} \\ S_{yu} & S_{yy} \end{bmatrix} = \begin{bmatrix} H_{uf} \\ H_{yf} \end{bmatrix} \begin{bmatrix} Q_{up} & Q_{yp} \end{bmatrix}.$$

Predictive controller basis

Noisy past data: $y(k) = y(k) + v(k), \quad v(k) \sim \mathcal{N}(0, \Sigma_v).$

A solution algorithm

1. Calculate x_u via $x_u = L_{up}^{-1} u_p.$
2. Estimate $\hat{x}_y$ via $\hat{x}_y = (L_{yp}^T \Sigma_v^{-1} L_{yp})^{-1} L_{yp}^T \Sigma_v^{-1} (y_p - L_{yup} x_u). \quad (\text{LS estimate})$
3. Calculate $\hat{z}$ via, $\hat{z} = L_{uf}^{-1} (u_f - S_{uu} x_u - S_{uy} \hat{x}_y).$
4. Predict $\hat{y}_f$ via, $\hat{y}_f = S_{yu} x_u + S_{yy} \hat{x}_y + L_{yf} \hat{z}.$

Properties (noise-free modeling data)

- ▶ If $v = 0$ then $\hat{y}_f = y_f \quad (\text{correct})$
- ▶ $\hat{y}_f$ is unbiased.
- ▶ Covariance of $\hat{y}_f$ is $(S_{yy} + L_{yf} L_{uf}^{-1} S_{uy}) (L_{yp}^T \Sigma_v^{-1} L_{yp})^{-1} (S_{yy} + L_{yf} L_{uf}^{-1} S_{uy})^T.$
- ▶ $\hat{y}_f$ is linear in past data, u_p, y_p , and the variables x_u, x_y, z , and u_f .

Predictive controller

Control problem:

Given immediate past data vectors, u_p , y_p ,

$$\underset{u_f, y_f}{\text{minimise}} \quad (y_f - r)^T Q (y_f - r) + u_f^T R u_f$$

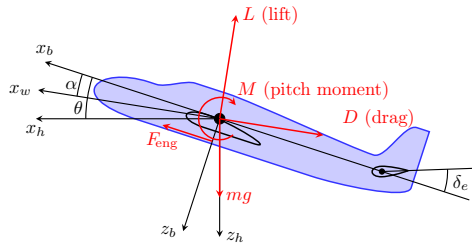
$$\text{subject to:} \quad y_f = X_1 u_p + X_2 y_p + X_3 u_f, \quad u_f \in \mathcal{U}, \quad y_f \in \mathcal{Y}.$$

This idea is essentially equivalent to subspace predictive control¹⁰.

¹⁰Favoreel, De Moor, Gevers, IFAC WC, 1999

A flight control example¹¹

Aircraft longitudinal dynamics: Boeing 747



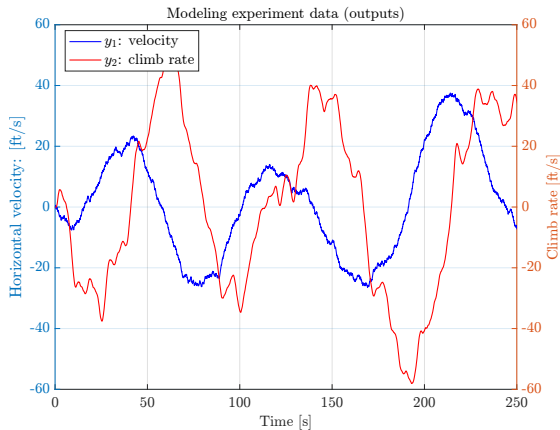
Nominal model is a 4 state system.

Actuation inputs:	u_1 : throttle
	u_2 : elevator angle
Measured outputs:	y_1 : velocity (body axis)
	y_2 : climb rate.

¹¹Verheijen, Breschi, Lazar, *Ann. Rev. Ctrl.*, 2023

A flight control example

Modeling data and control problem



Model data experiment: 2500 data points (250 seconds).

Past horizon: $T_p = 30$ (3 sec.)

Future horizon: $T_f = 20$ (2 sec.)

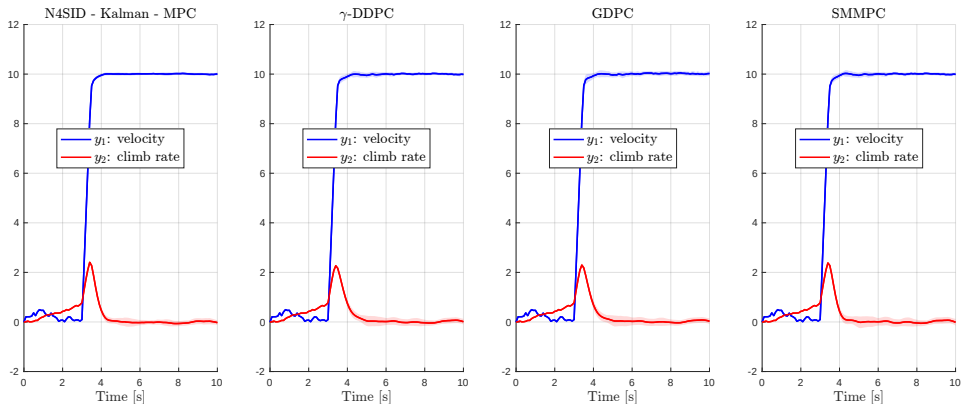
Noisy model data

30 Monte Carlo runs.

Control experiment:
10 ft/sec. velocity step.

A flight control example (n4sid corrected)

Step response simulation: comparison classical vs. data-driven

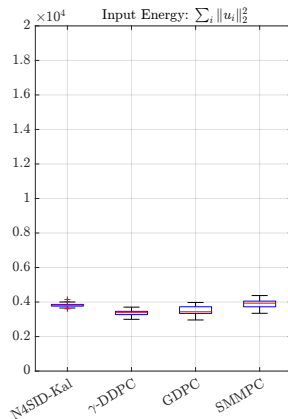
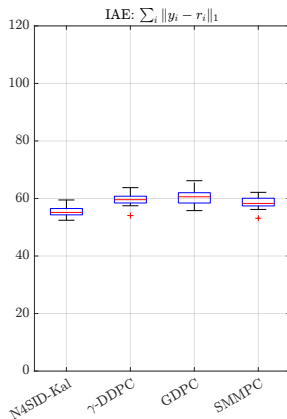
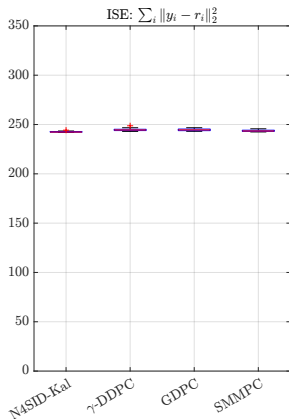


Classical methods: subspace identification (N4SID), Kalman filter, state-space MPC.

Data driven: Regularised data-driven (γ -DDPC, GDPC) and Signal Matrix Model based MPC.

A flight control example (n4sid corrected)

Step response simulation: Monte Carlo statistics



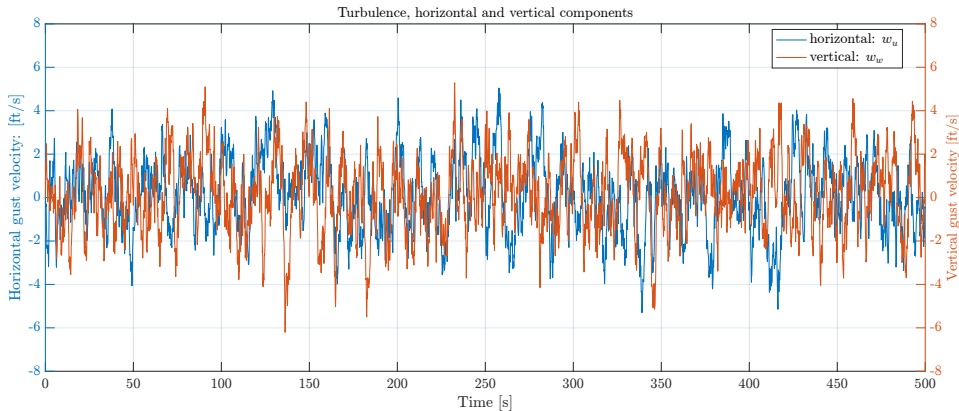
Classical methods: subspace identification (N4SID), Kalman filter, state-space MPC.

Data driven: Regularised data-driven (γ -DDPC, GDPC) and Signal Matrix Model based MPC.

What could possibly go wrong?

Turbulence!

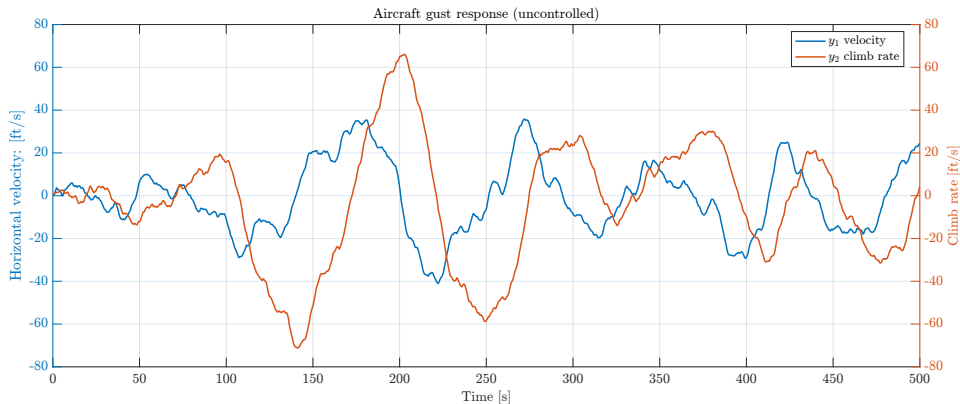
(Dryden gust model — 3 states)



How does the aircraft respond?

Process disturbance: turbulence

Uncontrolled turbulence response



Kalman filtering

Trade-off between disturbances and measurement noise

- ▶ The system is affected by unmeasured disturbance inputs.
- ▶ There is also measurement noise on the output.
- ▶ Estimate the state (optimal w.r.t. variance of the disturbances and variance of the measurement noise).

Evolution equation for x_u and x_y ?

$$\begin{bmatrix} u_p(k) \\ y_p(k) \end{bmatrix} = \begin{bmatrix} L_{up} & 0 \\ L_{yup} & L_{yp} \end{bmatrix} \begin{bmatrix} x_u(k) \\ x_y(k) \end{bmatrix}$$

next time step
 $\longrightarrow$

$$\begin{bmatrix} u_p(k+1) \\ y_p(k+1) \end{bmatrix} = \begin{bmatrix} L_{up} & 0 \\ L_{yup} & L_{yp} \end{bmatrix} \begin{bmatrix} x_u(k+1) \\ x_y(k+1) \end{bmatrix}$$

Kalman filtering

Time indexing notation

$$u_p(k) = \begin{bmatrix} u(k - T_p + 1) \\ \vdots \\ u(k) \end{bmatrix} \quad \text{and} \quad y_p(k) = \begin{bmatrix} y(k - T_p + 1) \\ \vdots \\ y(k) \end{bmatrix}.$$

A recursive predictor¹²

$$\underbrace{\begin{bmatrix} \hat{x}_u(k+1) \\ \hat{x}_y(k+1) \end{bmatrix}}_{x_{uy}(k+1)} = \underbrace{\begin{bmatrix} A_{uu} & 0 \\ A_{yu} & A_{yy} \end{bmatrix}}_{A_p} \underbrace{\begin{bmatrix} \hat{x}_u(k) \\ \hat{x}_y(k) \end{bmatrix}}_{x_{uy}(k)} + \underbrace{\begin{bmatrix} B_{uu} \\ B_{yu} \end{bmatrix}}_{B_p} u(k).$$

$$\hat{y}(k) = \underbrace{\begin{bmatrix} C_{yu} & C_{yy} \end{bmatrix}}_{C_p} \underbrace{\begin{bmatrix} \hat{x}_u(k) \\ \hat{x}_y(k) \end{bmatrix}}_{x_{uy}(k)} + \underbrace{\begin{bmatrix} D_{uu} \\ D_{yu} \end{bmatrix}}_{D_p} u(k).$$

¹²Smith, Abdalmoaty, Yin, *CDC*, 2024b.

Dynamic predictor model

Noise-free model data case

$$\begin{bmatrix} u_p \\ y_p \end{bmatrix} = \begin{bmatrix} L_{up} & 0 \\ L_{yup} & L_{yp} \end{bmatrix} \begin{bmatrix} x_u \\ x_y \end{bmatrix}$$

L_{yp} has n_x columns. We have an extended-state: $\begin{bmatrix} x_u \\ x_y \end{bmatrix} \in \mathcal{R}^{n_u T_p + n_x}$.

Noisy model data case

L_{yp} is square (but close to low rank); truncate or reduce column dimension.

Low rank (in the causality decomposition) is required for the recursive relationship.

Extended-state: $\begin{bmatrix} x_u \\ x_y \end{bmatrix} \in \mathcal{R}^{n_u T_p + n_y (T_p - 1)}$.

Kalman filtering

Kalman filtering procedure

Correct the prediction update by $K_{\text{kf}} (\hat{y}(k) - y_p(k))$.

Estimator equations:

$$\begin{aligned}\hat{x}_{uy}(k+1) &= A_p \hat{x}_{uy}(k) + B_{up} u(k) - K_{\text{kf}} (\hat{y}(k) - y_p(k)) \\ &= (A_p - K_{\text{kf}} C_p) \hat{x}_{uy}(k) + (B_{up} - K_{\text{kf}} D_p) u(k) + K_{\text{kf}} y_p(k) \\ \hat{y}(k) &= C_p \hat{x}_{uy}(k) + D_p u(k).\end{aligned}$$

Estimation error dynamics

$$e_{x_{uy}}(k+1) = (A_p - K_{\text{kf}} C_p) e_{x_{uy}}(k) + B_{wp} w(k) + K_{\text{kf}} v(k).$$

Amenable to standard Kalman filter design methods.

Output prediction

Output prediction using $\hat{x}_u, \hat{x}_y$ (effect of w_p may be able to be captured in $\hat{x}_u$)

Estimated future output:

$$\hat{y}_f = E_{x_u} \hat{x}_u + E_{x_y} \hat{x}_y + E_{uf} u_f,$$

where $E_{x_u} = (S_{yu} - L_{yf}L_{uf}^{-1}S_{uu})$, $E_{x_y} = (S_{yy} - L_{yf}L_{uf}^{-1}S_{uy})$, and $E_{uf} = L_{yf}L_{uf}^{-1}$.

Signal Matrix Model–Kalman predictive control

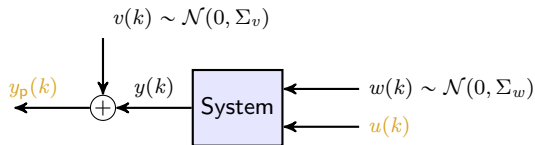
Certainty equivalence control objective:

$$\underset{u_f, y_f}{\text{minimise}} \quad (y_f - r)^T Q (y_f - r) + u_f^T R u_f^T$$

$$\text{subject to: } y_f = E_{x_u} \hat{x}_u + E_{x_y} \hat{x}_y + E_{uf} u_f, \quad u_f \in \mathcal{U}, \quad y_f \in \mathcal{Y}.$$

Exploiting dynamics in the process disturbance

Process noise inputs (a (u, w, y) trajectory is available)



Include w in the Signal Matrix Model

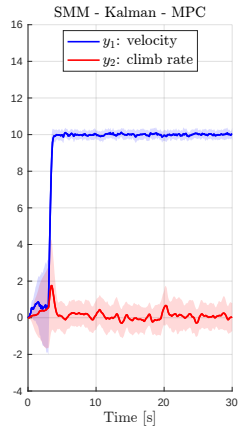
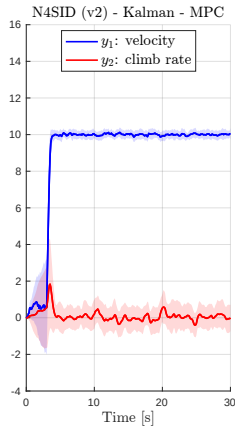
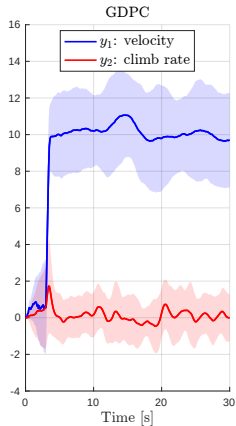
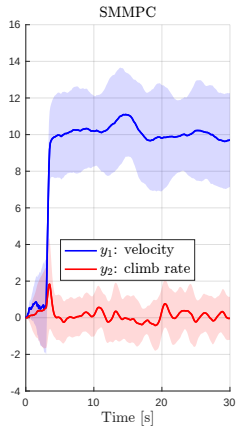
$$\begin{bmatrix} u_p \\ w_p \\ y_p \\ u_f \\ y_f \end{bmatrix} = \begin{bmatrix} L_{up} & 0 & 0 & | & 0 \\ L_{wup} & L_{wp} & 0 & | & 0 \\ L_{yup} & L_{ywp} & L_{yp} & | & 0 \\ \hline S_{uu} & S_{uw} & S_{uy} & | & L_{uf} \\ S_{yu} & S_{yw} & S_{yy} & | & L_{yf} \end{bmatrix} \begin{bmatrix} x_u \\ x_w \\ x_y \\ \hline z \end{bmatrix}$$

Reduced order estimate: $\hat{x}_w$ and $\hat{x}_y$.

Prediction for MPC: $y_f = E_{up} u_p + E_{x_w} \hat{x}_w + E_{x_y} \hat{x}_y + E_{uf} u_f$.

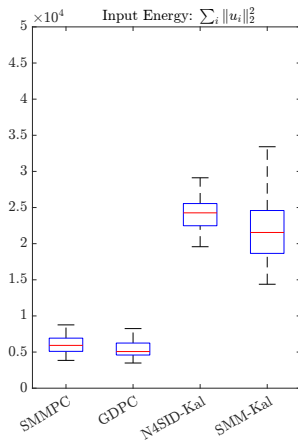
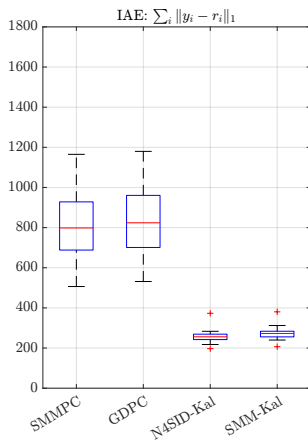
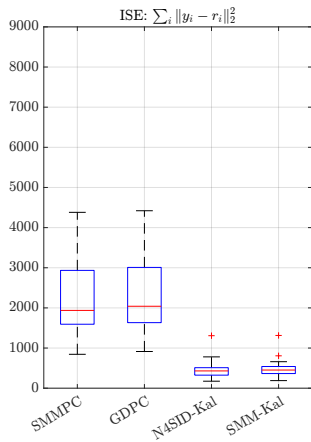
A flight control example (n4sid corrected)

Gust and step command response



A flight control example (n4sid corrected)

Gust and step command response: Monte Carlo statistics



Unmeasured disturbances in the model data

Process disturbances

1. The model data is corrupted by a (potentially dynamic) process disturbance, w .

The disturbance signal w is available for constructing the SMM.

In this case estimate $\hat{x}_w$ and $\hat{x}_y$ from u_p , y_p ; calculate x_u from u_p .

This is generally the case when the model data is generated by a simulation.

Dynamics/time correlation in w is now included in the SMM “state,” $\hat{x}_w$.

2. Both the model data and operational data are corrupted by an unmeasured dynamic process disturbance, w .

This is the more common case in dealing with measured data for the SMM.

SMM-based Kalman filtering for unmeasured process disturbances

State space representation

$$\begin{aligned}x(k+1) &= Ax(k) + B_u u(k) + B_w \mathbf{w}(k), & \mathbf{w}(k) &\sim \mathcal{N}(0, \Sigma_w) \\y(k) &= Cx(k) + Du(k) + \mathbf{v}(k), & \mathbf{v}(k) &\sim \mathcal{N}(0, \Sigma_v)\end{aligned}$$

Innovations representation¹³ (also used in subspace identification¹⁴)

This formulation is equivalent in terms of the variance of the error in estimating $\hat{\mathbf{y}}(k)$.

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= A \hat{\mathbf{x}}(k) + B_u u(k) + K_{\text{kf}} \mathbf{e}(k), & \mathbf{e}(k) &\sim \mathcal{N}(0, \Sigma_e) \\y(k) &= C \hat{\mathbf{x}}(k) + D u(k) + \mathbf{e}(k).\end{aligned}$$

Note that $\mathbf{e}(k) \in \mathcal{R}^{n_y}$ is white noise.

¹³See for example: Kailath, Sayed, Hassibi, 2000

¹⁴P. Van Overschee and B. De Moor, 2012

SMM-based Kalman filtering for unmeasured process disturbances

Standard Kalman filter

$$\hat{x}(k+1) = A \hat{x}(k) + B_u u(k) + K_{\text{kf}}(y(k) - \hat{y}(k))$$

$$\hat{y}(k) = C \hat{x}(k) + D u(k).$$

To implement the Kalman filter we rewrite this as,

$$\hat{x}(k+1) = (A - K_{\text{kf}}C) \hat{x}(k) + (B_u - K_{\text{kf}}D) u(k) + K_{\text{kf}} y(k)$$

$$\hat{y}(k) = C \hat{x}(k) + D u(k).$$

Kalman filter design (known system)

Optimal case: $K_{\text{kf}} = PC^T(CPC^T + \Sigma_v)^{-1}$,

with P satisfying $P = APA^T + \Sigma_w - APC^T(CPC^T + R)^{-1}CPA^T$.

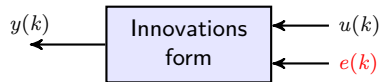
SMM-based Kalman filtering for unmeasured process disturbances

Transforming to the innovations form

Define $e(k) = y(k) - \hat{y}(k) = y(k) - C\hat{x}(k) - Du(k)$ to get

$$\hat{x}(k+1) = A\hat{x}(k) + B_u u(k) + K_{kf} e(k)$$

$$y(k) = C\hat{x}(k) + Du(k) + e(k).$$



Causal invertibility (“whitening” filter)

The advantage of replacing $w(k)$ and $v(k)$ with $e(k)$ is that $e(k)$ is causally related to the measured input-output data.

$$\hat{x}(k+1) = (A - K_{kf}C)\hat{x}(k) + (B_u - K_{kf}D)u(k) + K_{kf}y(k)$$

$$e(k) = y(k) - C\hat{x}(k) - Du(k).$$



SMM-based Kalman filtering for unmeasured process disturbances ¹⁷

Reconstructing the innovations sequence, $e(k)$

$$e(l) = C(A - K_{kf}C)^{l-1}x(1) + C \sum_{k=1}^l (A - K_{kf}C)^{l-k-1} (B_w w(k) - K_{kf}v(k))$$



We don't know $x(1)$, or the signals $w(k)$, $v(k)$, but we can simulate $e(k)$ from an SMM using $u(k)$ and $y(k)$.

The SMM simulation of the innovations sequence will have a transient error (from unknown $x(1)$). If Σ_v is small w.r.t. Σ_w , the transient decays quickly.

Closed-loop subspace identification uses a similar trick^{15 16}.

¹⁵Qin & Ljung, *IFAC Proc.*, 2003

¹⁶Chuiso, *Automatica*, 2007

¹⁷M. Abdalmoaty and R.S. Smith, *ECC*, 2026.

SMM-based Kalman filtering for unmeasured process disturbances

Estimating the innovations sequence, $e(k)$

Use an SMM to simulate the whitening filter.



$$\begin{bmatrix} \frac{H_L \left(\begin{bmatrix} u(1:L+N) \\ y(1:L+N) \end{bmatrix} \right)}{H_1(y(L+1:L+N+1))} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} u(1) \\ y(1) \end{bmatrix} & \cdots & \begin{bmatrix} u(N) \\ y(N) \end{bmatrix} \\ \vdots & & \vdots \\ \begin{bmatrix} u(L) \\ y(L) \end{bmatrix} & \cdots & \begin{bmatrix} u(L+N) \\ y(L+N) \end{bmatrix} \\ \hline y(L+1) & \cdots & y(L+N+1) \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{12} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}.$$

Simulating $\begin{bmatrix} u(1:L) \\ y(1:L) \end{bmatrix} = 0$ gives $y(L+1:L+N+1) \approx e(L+1:L+N+1)$.

So $\hat{e}(L+1:L+N+1) = y(L+1:L+N+1) = L_{22}Q_2^T$.

Innovations-based SMM Kalman predictor

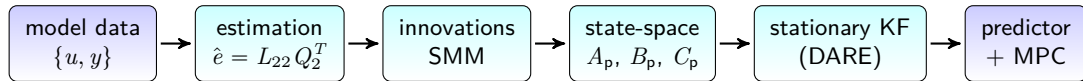
SMM driven by innovations

$$\begin{bmatrix} u_p \\ e_p \\ y_p \\ u_f \\ y_f \end{bmatrix} = \left[\begin{array}{ccc|c} L_{up} & 0 & 0 & 0 \\ L_{ep} & L_{ep} & 0 & 0 \\ L_{yp} & L_{yp} & L_{yp} & 0 \\ \hline S_{uu} & S_{ue} & S_{uy} & L_{uf} \\ S_{yu} & S_{ye} & S_{yy} & L_{yf} \end{array} \right] \begin{bmatrix} x_u \\ x_e \\ x_y \\ z \end{bmatrix}$$

Reduced order estimate: $\hat{x}_e$ and $\hat{x}_y$.

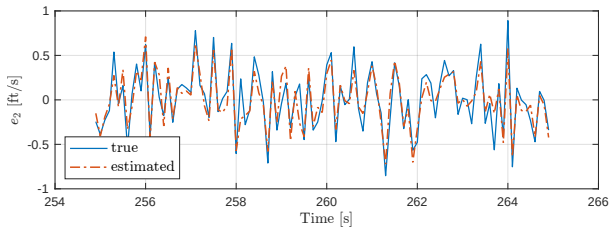
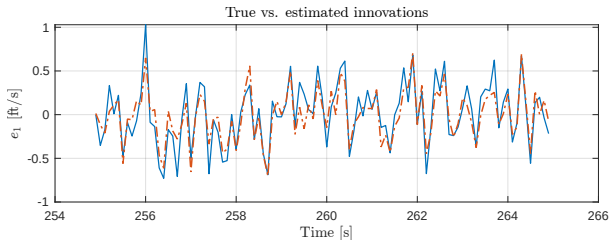
Prediction for MPC: $y_f = E_{up}u_p + E_{x_e}\hat{x}_e + E_{x_y}\hat{x}_y + E_{uf}u_f$.

Complete innovations-based SMM prediction control algorithm



Flight control example

Reconstructing the innovations sequence, $e(k)$



Boeing 747 example

$L = 150$

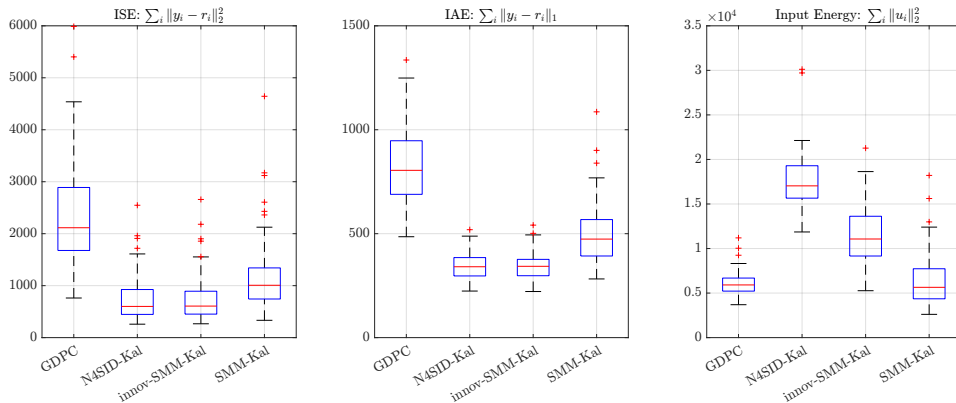
(chosen by cross-validation)

Last 10 seconds of $\hat{e}(k)$ and $e(k)$ are shown.

Effectively estimates the effect of unknown noise, $v(k)$, and process disturbance, $w(k)$.

Flight control example

Closed-loop predictive performance (100 Monte-Carlo runs)



Note: innov-SMM-Kal outperforms SMM-Kal even though w is not known.

Summary and comments

Signal Matrix Model formulations

- ▶ Integrates easily into a predictive control scheme.
- ▶ Causal decoupling removes the need for regularisation
- ▶ SMM and data-driven predictive control needs **all** input signals to work well.

Process noise in data-driven control

- ▶ Unmeasured process noise (and/or time-correlated measurement noise) severely degrades SMM predictions.
- ▶ Estimation of the dynamic effect of process and measurement noise needed.

Kalman filtering and innovations

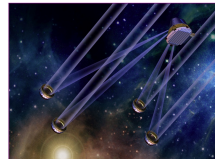
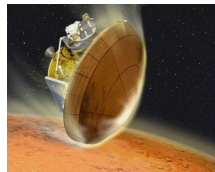
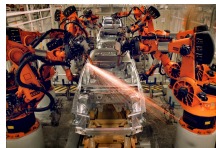
- ▶ Ability to track the effect of unknown/uncertain process disturbances.
- ▶ Kalman filter averages noise over all past data (not just the horizon T_p).
- ▶ Estimating innovations recovers optimal trade-off between measurement and process noise.
- ▶ Process disturbance forecasts easily incorporated.

Predictive models: state-based vs. data-based

State-based model applications

- ▶ High economic value
- ▶ Safety critical
- ▶ Hierarchical or distributed
- ▶ Mass-produced

The physical state may have a meaning that makes it worth modeling and identifying.



Data-driven methods: further considerations

Benefits

- ▶ The predictive modeling process is simplified.
- ▶ Faster to implement.
- ▶ Potentially cost effective for low-value heterogeneous subsystems.

Drawbacks

- ▶ Measurement data is messy and unreliable.
- ▶ Experimental and identification considerations subsumed in getting the data.
- ▶ Operational data is poorly suited to characterising system behaviours (insufficiently exciting).
- ▶ Difficult to incorporate physical knowledge.
- ▶ Fits poorly into a complex system hierarchy. (lose meaning associated with model parameters).

Bibliography

Data driven Hankel matrix formulations

1. J.C. Willems, P. Rapisarda, I. Markovsky, and B.L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, **54**(4), pp. 325–329, 2005.
2. C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.
3. I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. Journal of Control*, **81**(12), pp. 1946–1959, 2008.
4. J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," *Proc. European Control Conf.*, pp. 307–312, 2019.
5. V. Breschi, A. Chiuso, and S. Formentin, "Data-driven predictive control in a stochastic setting: a unified framework," *Automatica*, **152**, p. 110961, 2023.
6. M. Lazar and P. Verheijen, "Generalized data-driven predictive control: Merging subspace and Hankel predictors," *Mathematics*, **11**, p. 2216, 2023.
7. P. Mattsson and T. Schön, "On the regularization in DeePC," *IFAC PapersOnLine*, **56**(2), pp. 625–631, 2023.

Bibliography

Data driven Hankel matrix formulations (continued)

8. M. Yin, A. Iannelli, and R.S. Smith, "Maximum likelihood estimation in data-driven modeling and control," *IEEE Trans. Automatic Control*, **68**(1), pp. 317–328, 2023.
9. R.S. Smith, M. Abdalmoaty, and M. Yin, [2024a] "Optimal data-driven prediction and predictive control using signal matrix models," *IEEE Conf. Decision & Control (CDC)*, pp. 6767–6773, 2024. [Complete code at doi: 10.3929/ethz-b-000663423](https://doi.org/10.3929/ethz-b-000663423).
11. P. Verheijen, V. Breschi, and M. Lazar, "Handbook of linear data-driven predictive control: Theory, implementation and design," *Annual Reviews in Control*, **56**, p. 100914, 2023. ¹⁸
12. R. S. Smith, M. Abdalmoaty, and M. Yin, [2024b] "Data-driven formulation of the Kalman filter and its application to predictive control," *Conf. Decision & Control (CDC)*, pp. 2633–2639, 2024. [Complete code at doi: 10.3929/ethz-b-000693730](https://doi.org/10.3929/ethz-b-000693730).

¹⁸The example in this paper uses an `n4sid` default which is poorly suited to the problem making the `n4sid`-Kalman-MPC performance much worse than it should be. The same problem occurs in [9] and [12]. It is corrected and updated in the code associated with [9] and [12].

Bibliography

Subspace predictive control

10. W. Favoreel, B. De Moor, and M. Gevers, “SPC: Subspace predictive control,” *Proc. IFAC World Congress*, pp. 4004–4009, 1999.

Estimation and Innovation formulations

13. T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.
14. P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.
15. S.J. Qin and L. Ljung, “Closed-loop subspace identification with innovation estimation,” *IFAC Proceedings Volumes*, **36**(16), pp. 861–866, 2003.
16. A. Chiuso, “The role of vector autoregressive modeling in predictor-based subspace identification,” *Automatica*, **43**(6), pp. 1034–1048, 2007.
17. M. Abdalmoaty and R.S. Smith, “An Innovations-Based Data-Driven Kalman Predictor for Predictive Control,” *Proc. European Control Conf.*, pp. 56–61, 2026. [Complete code at doi: 10.3929/ethz-c-000798590](https://doi.org/10.3929/ethz-c-000798590).