2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Data-driven formulation of the Kalman filter and its Application to Predictive Control

Roy S. Smith, Mohamed Abdalmoaty, & Mingzhou Yin

Abstract— **Data-driven methods for predictive control rely on input-output data to give a Hankel matrix representation of the space of trajectories. They are poorly suited to situations where both process noise and measurement noise dominate the behaviour whereas Kalman filters optimally estimate system states in this scenario. We derive a data-driven Kalman filter formulation based on the dynamic evolution of Hankel matrix output predictions. This leads to an extended state space model that describes the evolution of both the future inputs and outputs. By applying measurement feedback one arrives at a Kalman filter for the system. The Kalman filter design is performed purely on the basis of the input and output signals and without the need for a specific state-space representation. A benchmark simulation illustrates that the resulting prediction-based control significantly out-performs predictive controllers based on current data-driven methods.**

I. Introduction

Data-driven predictive control uses a past data characterisation of a system's input-output dynamics to determine a future control input sequence. Most of the current work is based on the Willems' fundamental lemma [1], [2] which provides a linear characterisation of all input-output sequences compatible with the system dynamics. This has been effectively used to develop predictive control methods using past data sequences. Examples of such methods are DeePC [3], γ-DDPC [4], [5], and GDPC [6]. See [7] for a comparative review. One aspect of these methods is the use of regularisation to trade-off between the matching of initial conditions and the control performance. Subspace Predictive Control (SPC) [8] can be derived in terms of the same Hankel matrices but does not use regularisation for the control design.

In the case where the past data is corrupted by noise, the characterisation of the system dynamics becomes inaccurate resulting in regression formulations that have an error-in-variables structure. Maximum likelihood approaches to predictive control in this case are considered in [9], [10].

Data-driven predictive control methods are not "model-free", but rather use matrices formed from past data to parameterise the model. We refer to this class of linear time-invariant (LTI) models as Signal Matrix Models (SMM).

This paper develops a method of creating Kalman filters [11] directly from past data using SMMs. The use of Kalman filters allows us to extend the domain of current data-driven predictive control methods to the inclusion of unknown stochastically driven inputs. In this work we consider that noise-free past data is used in the SMM. This is of course not true in practice and this assumption amounts to the use of an inaccurate model of the dynamics. If the noise effects in the past data are small (or made small by good experiment and input design) the SMM model will be a close approximation to the dynamics. For sample complexity guarantees in a similar framework see [12]. In the noisy case one can "clean" the data via Hankel matrix de-noising techniques [13]. Our theoretical work applies to the noise-free model data case, but we carry out a simulation study in Section IV using noisy model data.

The characterisation of data-driven predictive control in terms of Kalman filtering has been considered in [14]. This work uses averaged past trajectories to formulate a extended Kalman filter form of prediction update. The EKF formulation is nonlinear making it difficult to characterise and design. Another approach using maximum likelihood is considered in [15], where the state-space is prespecified in terms of an initial state but the A, B, C, and D matrices themselves are unknown. A more comprehensive and similarly motivated approach is given in [16] where a time-varying Kalman filter is used in the context of a stochastic data-driven predictive control problem. This work considers a more stochastic formulation with noise in the past data SMM matrices as well as the current measurements. The associated predictive control problem considers stochastic chance constraints and uses an expected cost formulation.

The approach in the current paper is more deterministic in the assumptions on the SMM. Stochasticity in our problem arises only from measurement and process noise. In contrast to [15] we avoid the definition of the state-space itself. Any important internal variables should be able to be measured (simulated) or reconstructed from input-output data, so that they can be predicted, monitored, or potentially constrained in a predictive control problem.

Our method is based on a reformulation of the signal constraints in Willems' lemma that gives a decoupling of the initial condition matching and predictive control parts of the problem [17]. We have developed a similar Kalman filter for stochastic predictive control problems in [18].

A. Notation

Gaussian distributions with mean μ and covariance Σ are denoted by $\mathscr{N}(\mu,\Sigma)$. The expectation of a random vector x is $\mathscr{E}\{x\}$ and its covariance matrix is $\operatorname{cov}(x)$. A matrix X

This work was supported by the Swiss National Science Foundation NCCR Automation, grant agreement 51NF40_180545.

R. Smith and M. Abdalmoaty are with the Automatic Control Laboratory (IfA), ETH Zürich, Physikstrasse 3, 8092 Zürich, Switzerland, (e-mail: {rsmith,mabdalmoaty}@control.ee.ethz.ch. M. Yin was with IfA, ETH and is now with the Institute of Automatic Control, Leibniz University Hannover, Germany (email: yin@irt.uni-hannover.de)

has a range space range(X). The Kronecker product between matrices X and Y is denoted by $X \otimes Y$. Symmetric matrices of dimension n are denoted $\mathbb{S}^n$, with positive definite (positive semidefinite) matrices denoted by $\mathbb{S}^n_{++}$ ($\mathbb{S}^n_{+}$). The weighted l_2-norm of a vector $x \in \mathbb{R}^n$ is $\|x\|_P = (x^T P x)^{1/2}$ with $P \in \mathbb{S}^n_{++}$.

II. Background and problem formulation

A. Model assumptions

We assume that the system to be estimated and controlled is a finite (n_x) dimensional discrete-time, strictly causal, linear time-invariant system, G, represented by a state-space formulation with input, $u(k) \in \mathbb{R}^{n_u}$, and output, $y(k) \in \mathbb{R}^{n_y}$,

$$x(k+1) = Ax(k) + B_u u(k) + B_w w(k), \quad (1)$$
$$y(k) = Cx(k) + v(k), \quad (2)$$

where $w(k) \in \mathbb{R}^{n_w}$, and $v(k) \in \mathbb{R}^{n_y}$, are process (or disturbance) and measurement noises respectively. We assume the noises to be independent and drawn from normal distributions of known variances,

$$w(k) \sim \mathcal{N}(0, \Sigma_w), \quad v(k) \sim \mathcal{N}(0, \Sigma_v), \quad (3)$$

In contrast to the state-space representation above, our characterisation of G is based on the Willems' lemma. We have a length-K, trajectory $\{u^{\mathrm{d}}(k), w^{\mathrm{d}}(k), y^{\mathrm{d}}(k)\}$, $k = 1, \ldots, K$. The d superscript denotes the data used for characterising the sytem. In the SMM construction step we assume that $w^{\mathrm{d}}(k)$ is available. In the online application of the Kalman filter the disturbance signal, $w(k)$, is not available.

We consider a contiguous, length-T, sequence of inputs and outputs. This is partitioned into T_{p} immediate past time points and T_{f} future time points, with $T = T_{\mathrm{p}} + T_{\mathrm{f}}$. The sequence of immediate past inputs and outputs are considered as a vector and denoted by,

$$u_{\mathrm{p}}(k) = \begin{bmatrix} u(k-T_{\mathrm{p}}) \\ \vdots \\ u(k-1) \end{bmatrix} \text{ and } y_{\mathrm{p}}(k) = \begin{bmatrix} y(k-T_{\mathrm{p}}+1) \\ \vdots \\ y(k) \end{bmatrix}. \quad (4)$$

The immediate future inputs and outputs are then

$$u_{\mathrm{f}}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+T_{\mathrm{f}}-1) \end{bmatrix} \text{ and } y_{\mathrm{f}}(k) = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+T_{\mathrm{f}}) \end{bmatrix}. \quad (5)$$

Note that in forming these signal vectors y_{p} and y_{f} are one time step advanced compared to the u_{p} and u_{f}.

The past horizon T_{p} must be large enough to completely characterise the effects of all past inputs on the future input-output mapping. This requires that the input, u^{d}, is persistency exciting of order at least $T_{\mathrm{p}}(n_u + n_w) + n_x$ and that $n_y T_{\mathrm{p}} \geq n_x$. We will also assume that the data-length, K, satisfies $K \geq 2T(n_y + n_u + n_w) + T$. These conditions can be weakened but are easily satisfied and provide adequate data for Kalman filtering and predictive control.

B. Willems' lemma

The basis for most data-driven predictive control is found in Willems' lemma. This will be stated here in terms of a generic input u^{d}, but in the Kalman filtering application we will actually include w^{d} in this input vector.

We define Hankel matrices, of dimension $(n_u T) \times M$ and dimension $(n_y T) \times M$,

$$H_u = \begin{bmatrix} u^{\mathrm{d}}(1) & u^{\mathrm{d}}(2) & \cdots & u^{\mathrm{d}}(M) \\ u^{\mathrm{d}}(2) & u^{\mathrm{d}}(3) & \cdots & u^{\mathrm{d}}(M+1) \\ \vdots & \vdots & & \vdots \\ u^{\mathrm{d}}(T) & u^{\mathrm{d}}(T+1) & \cdots & u^{\mathrm{d}}(M+T-1) \end{bmatrix} \in \mathbb{R}^{n_u T \times M},$$

and

$$H_y = \begin{bmatrix} y^{\mathrm{d}}(2) & y^{\mathrm{d}}(3) & \cdots & y^{\mathrm{d}}(M+1) \\ y^{\mathrm{d}}(3) & y^{\mathrm{d}}(4) & \cdots & y^{\mathrm{d}}(M+2) \\ \vdots & \vdots & & \vdots \\ y^{\mathrm{d}}(T+1) & y^{\mathrm{d}}(T+2) & \cdots & y^{\mathrm{d}}(M+T) \end{bmatrix} \in \mathbb{R}^{n_y T \times M}.$$

If all of the K available input-output signal data are used then $M = K - T$. The matrix H_u has full row rank by the persistency of excitation assumption. Generically H_y also has full row rank (equal to $n_y T$).

Lemma 1 (Willems' Fundamental Lemma): Under the above assumptions on data length and persistency of excitation, the length-T input-output pair (u,y) is a nominal trajectory of the system G (i.e. $w(k) = 0$ and $v(k) = 0$), iff there exists $g \in \mathbb{R}^M$ such that,

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} H_u \\ H_y \end{bmatrix} g. \quad (6)$$

See Theorem 1 in [1] and Lemma 2 in [2]. See [19] for the necessity of the persistency of excitation assumption. By Willems' lemma, the stacking of these matrices has rank equal to $n_u T + n_x$. Furthermore the if and only if nature of the result characterises all of the possible system responses in terms of $g \in \mathbb{R}^M$.

To apply the characterisation in (6) to estimation and control we partition the H_u and H_y SMM matrices commensurately with the past and future time windows given in (4) and (5),

$$H_u = \left[\begin{array}{c} H_{u\mathrm{p}} \\ \hline H_{u\mathrm{f}} \end{array}\right], \text{ with } H_{u\mathrm{p}} \in \mathbb{R}^{(n_u T_{\mathrm{p}}) \times M}, H_{u\mathrm{f}} \in \mathbb{R}^{(n_u T_{\mathrm{f}}) \times M},$$

and

$$H_y = \left[\begin{array}{c} H_{y\mathrm{p}} \\ \hline H_{y\mathrm{f}} \end{array}\right], \text{ with } H_{y\mathrm{p}} \in \mathbb{R}^{(n_y T_{\mathrm{p}}) \times M}, H_{y\mathrm{f}} \in \mathbb{R}^{(n_y T_{\mathrm{f}}) \times M}.$$

The assumptions in Section II-A imply that the rank and excitation assumptions in Lemma 1 also apply to H_u.

This allows us to express (6) in the form,

$$\begin{bmatrix} u_{\mathrm{p}} \\ y_{\mathrm{p}} \\ u_{\mathrm{f}} \\ y_{\mathrm{f}} \end{bmatrix} = \begin{bmatrix} H_{u\mathrm{p}} \\ H_{y\mathrm{p}} \\ H_{u\mathrm{f}} \\ H_{y\mathrm{f}} \end{bmatrix} g \quad (7)$$

This characterisation is directly used to specify the dynamics in DeePC. The γ-DDPC and GDPC methods use modified versions with a lower dimensional representation than that parametrised by $g \in \mathbb{R}^M$. This is also true of our work.

C. A minimal data-driven characterisation

We introduce a characterisation that is equivalent to (7),

$$\begin{bmatrix} u_\mathrm{p} \\ y_\mathrm{p} \\ u_\mathrm{f} \\ y_\mathrm{f} \end{bmatrix} = \begin{bmatrix} L_{u\mathrm{p}} & 0 & 0 \\ L_{yu\mathrm{p}} & L_{y\mathrm{p}} & 0 \\ S_{uu} & S_{uy} & L_{u\mathrm{f}} \\ S_{yu} & S_{yy} & L_{yu\mathrm{f}} \end{bmatrix} \begin{bmatrix} x_u \\ x_y \\ z \end{bmatrix}, \qquad (8)$$

where $x_u \in \mathbb{R}^{n_u T_\mathrm{p}}$, $x_y \in \mathbb{R}^{n_x}$, and $z \in \mathbb{R}^{n_u T_\mathrm{f}}$. A complete derivation is given in [17]. The matrix in (8) is lower triangular and full column rank. It also holds that $L_{u\mathrm{p}}$, $L_{y\mathrm{p}}$, and $L_{u\mathrm{f}}$ are also lower triangular matrices. The basis for this derivation is the use of an LQ decomposition of the past matrices and an LQ decomposition the future matrices, constrained to be in the null-space of the past matrices.

The major benefit of this characterisation is that for all u_p, y_p sequences generated by the system, (8) specifies unique x_u, x_y vectors. Furthermore, as $L_{u\mathrm{f}}$ is full rank (see [17]), z specifies all of the available degrees of freedom in using a future input, u_f, to produce a future output, y_f. The contribution of the past input and output to the future signals is captured in the S matrices.

The separation of the SMM into x_u and x_y (determined by u_p and y_p) and z (determining the future output y_f that can be controlled by a future input u_f) is exploited in [17] to separate the estimation of initial conditions from the predictive control of the future trajectories. Here we will use it to develop a Kalman filter.

III. KALMAN FILTERING

Data-driven predictive control essentially solves memoryless problems at each time-step. Given immediate past sequences $u_\mathrm{p}(k)$ and $y_\mathrm{p}(k)$, the optimal future input sequence, $u_\mathrm{f}(k)$, is found via optimisation. The calculation of $u_\mathrm{f}(k+1)$ (and the associated $y_\mathrm{f}(k+1)$) at the next time step does not make any use of the previous solution. The uses information from only the previous length-T_p input-output measurements. A Kalman filter on the other hand uses information from the entire past history in predicting future outputs. Another, perhaps more valuable, advantage is that the effect of any unmeasured process noise $w(k)$ on the output is also estimated.

A. Derivation of a state-space predictor

The derivation starts from the observation that at the $k+1$ time step the variables $x_u(k+1)$ and $x_y(k+1)$ can be calculated in terms of $x_u(k)$, $x_y(k)$, and the input, $u(k)$. More precisely, we will derive a representation of the form,

$$x_u(k+1) = A_{uu}\, x_u(k) + A_{uy}\, x_y(k) + B_{uu}\, u(k), \qquad (9)$$

and

$$x_y(k+1) = A_{yu}\, x_u(k) + A_{yy}\, x_y(k) + B_{yu}\, u(k). \qquad (10)$$

To derive the matrices in (9) and (10) we define a matrix $Z_{u\mathrm{p}} \in \mathbb{R}^{(T_\mathrm{p} n_u)\times(T_\mathrm{p} n_u)}$ that maps the vector of past inputs, u_p, to its form one time step into the future,

$$\begin{bmatrix} u(k-T_\mathrm{p}+1) \\ \vdots \\ u(k-1) \\ 0 \end{bmatrix} = Z_{u\mathrm{p}} \begin{bmatrix} u(k-T_\mathrm{p}) \\ \vdots \\ u(k-2) \\ u(k-1) \end{bmatrix}$$

The matrix $Z_{y\mathrm{p}} \in \mathbb{R}^{(T_\mathrm{p} n_y)\times(T_\mathrm{p} n_y)}$ performs the analogous shift for the past output vectors, y_p. Define a matrix $\Pi_{u,k}$ that maps $u(k)$ onto the most recent part of the past input vector $u_\mathrm{p}(k)$.

$$\Pi_{u,k} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \otimes I_{n_u}.$$

This gives the relationship,

$$u_\mathrm{p}(k+1) = Z_{u\mathrm{p}}\, u_\mathrm{p}(k) + \Pi_{u,k}\, u(k). \qquad (11)$$

Substituting $u_\mathrm{p} = L_{u\mathrm{p}}\, x_u$ from the first row of (8) leads to,

$$A_{uu} = L_{u\mathrm{p}}^{-1} Z_{u\mathrm{p}} L_{u\mathrm{p}}, \quad A_{uy} = 0, \quad \text{and} \quad B_{uu} = L_{u\mathrm{p}}^{-1} \Pi_{u,k}.$$

To derive the matrices in (10) we define a matrix $\Pi_{y,k}$ analogously to $\Pi_{u,k}$,

$$\Pi_{y,k} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \otimes I_{n_y}.$$

We can now express $y_\mathrm{p}(k)$ in the form,

$$y_\mathrm{p}(k+1) = Z_{y\mathrm{p}}\, y_\mathrm{p}(k) + \Pi_{y,k}\, y(k+1). \qquad (12)$$

Substituting the second row of (8) into (12) gives,

$$L_{yu\mathrm{p}}\, x_u(k+1) + L_{y\mathrm{p}}\, x_y(k+1) = Z_{y\mathrm{p}}\, y_\mathrm{p}(k) + \Pi_{y,k}\, y(k+1).$$

We now substitute (9) for $x_u(k)$ in the above and rearrange to get,

$$L_{y\mathrm{p}}\, x_y(k+1) = \left(Z_{y\mathrm{p}} L_{yu\mathrm{p}} - L_{yu\mathrm{p}} A_{uu}\right) x_u(k) + Z_{y\mathrm{p}} L_{y\mathrm{p}}\, x_y(k) - L_{yu\mathrm{p}} B_{uu}\, u(k) + \Pi_{y,k}\, y(k+1). \qquad (13)$$

Define a matrix, $\Pi_{y,\mathrm{p}}$ that truncates the vector y_p to its earliest $T_\mathrm{p}-1$ components,

$$\Pi_{y,\mathrm{p}} = \begin{bmatrix} I_{T_\mathrm{p}-1} & 0 \end{bmatrix} \otimes I_{n_y}.$$

We now multiply (13) by $\Pi_{y,\mathrm{p}}$ and exploit the fact that $\Pi_{y,\mathrm{p}} \Pi_{y,k} = 0$ to get

$$\Pi_{y,\mathrm{p}} L_{y\mathrm{p}}\, x_y(k+1) = \Pi_{y,\mathrm{p}} \left(Z_{y\mathrm{p}} L_{yu\mathrm{p}} - L_{yu\mathrm{p}} A_{uu}\right) x_u(k) + \Pi_{y,\mathrm{p}} Z_{y\mathrm{p}} L_{y\mathrm{p}}\, x_y(k) - \Pi_{y,\mathrm{p}} L_{yu\mathrm{p}} B_{uu}\, u(k).$$

For notational convenience we define $\Psi_{y,\mathrm{p}} = \Pi_{y,\mathrm{p}} L_{y\mathrm{p}}$. As $\Psi_{y,\mathrm{p}}$ has full column rank, equal to n_x, the $x_y(k)$ update matrices in (10) are given by

$$A_{yu} = \left(\Psi_{y,\mathrm{p}}^T \Psi_{y,\mathrm{p}}\right)^{-1} \Psi_{y,\mathrm{p}}^T \left(\Pi_{y,\mathrm{p}} Z_{y\mathrm{p}} L_{yu\mathrm{p}} - \Pi_{y,\mathrm{p}} L_{yu\mathrm{p}} A_{uu}\right),$$

$$A_{yy} = \left(\Psi_{y,\mathrm{p}}^T \Psi_{y,\mathrm{p}}\right)^{-1} \Psi_{y,\mathrm{p}}^T \left(\Pi_{y,\mathrm{p}} Z_{y\mathrm{p}} L_{y\mathrm{p}}\right),$$

and

$$B_{yu} = \left(\Psi_{y,\mathrm{p}}^T \Psi_{y,\mathrm{p}}\right)^{-1} \Psi_{y,\mathrm{p}}^T \left(\Pi_{y,\mathrm{p}} L_{yu\mathrm{p}} B_{uu}\right).$$

We can express $y(k)$ in terms of the $x_u(k)$ and $x_y(k)$ vectors via,

$$y(k) = \Pi_{y,k}^T y_{\mathrm{p}}(k) = \Pi_{y,k}^T L_{yu\mathrm{p}} x_u(k) + \Pi_{y,k}^T L_{y\mathrm{p}} x_y(k).$$

Now define $C_{yu} = \Pi_{y,k}^T L_{yu\mathrm{p}}$, and $C_{yy} = \Pi_{y,k}^T L_{y\mathrm{p}}$. Denote the combined x_u, x_y vector as $x_{uy} = \begin{bmatrix} x_u^T & x_y^T \end{bmatrix}^T$. We can now summarise the model in state-space form,

$$x_{uy}(k+1) = A_{\mathrm{p}} x_{uy}(k) + B_{\mathrm{p}} u(k) \tag{14}$$

$$y(k) = C_{\mathrm{p}} x_{uy}(k), \tag{15}$$

where

$$A_{\mathrm{p}} = \begin{bmatrix} A_{uu} & 0 \\ A_{yu} & A_{yy} \end{bmatrix}, \quad B_{\mathrm{p}} = \begin{bmatrix} B_{uu} \\ B_{yu} \end{bmatrix}, \quad \text{and} \quad C_{\mathrm{p}} = \begin{bmatrix} C_{yu} & C_{yy} \end{bmatrix}.$$

Note that the state, x_{uy}, contains more information than required to simply determine $y(k)$. Using the last two rows of (8) one can also determine the input-output mapping T_{f} time steps into the future. This aspect will be used for model predictive control.

B. Kalman filter design

As (14) and (15) are in a standard state-space form all of the usual methods for Kalman filter design can be applied. One distinction from the standard case is that there are two inputs to the dynamics, $w(k) \in \mathbb{R}^{n_w}$ and $u(k) \in \mathbb{R}^{n_u}$. For the purposes of deriving the model we assume that the process disturbance sequence, $w^{\mathrm{d}}(k)$, is known. In practice this might come from historical measurements of past disturbances, or (as in the example) a disturbance rejection requirement on a simulation design model.

For the purposes of the model the known sequence $w^{\mathrm{d}}(k)$ is included as a component of H_u (and by extension $H_{u\mathrm{p}}$ and $H_{u\mathrm{f}}$) making $H_u \in \mathbb{R}^{(n_w+n_u)T \times M}$. Now by identifying the components associated with $u(k)$ and $w(k)$ we can partition the predictor B_{p} matrix in (14) into,

$$B_{\mathrm{p}} = \begin{bmatrix} B_{w\mathrm{p}} & B_{u\mathrm{p}} \end{bmatrix}.$$

This allows us to include w and v in the model of the extended system dynamics,

$$x_{uy}(k+1) = A_{\mathrm{p}} x_{uy}(k) + B_{w\mathrm{p}} w(k) + B_{u\mathrm{p}} u(k) \tag{16}$$

$$y_{\mathrm{meas}}(k) = C_{\mathrm{p}} x_{uy}(k) + v(k). \tag{17}$$

The unknown process noise $w(k)$ enters into the update equation for x_u meaning that both x_u and x_y need to be estimated.[1] Denote the estimate of x_{uy} by $\hat{x}_{uy}$ and the error in this estimate by, $e_{x_{uy}} = x_{uy} - \hat{x}_{uy}$.

Using a standard infinite horizon Kalman filter design, with Kalman gain K_{kf}, the state-space representation of the error dynamics (from inputs $w(k)$ and $v(k)$ to output $e_{x_{uy}}(k)$) is,

$$P_{\mathrm{err}}(z) = \left[\begin{array}{c|cc} A_{\mathrm{p}} - K_{\mathrm{kf}} C_{\mathrm{p}} & B_{w\mathrm{p}} \Sigma_w^{1/2} & K_{\mathrm{kf}} \Sigma_v^{1/2} \\ \hline I_{n_e} & 0 & 0 \end{array}\right].$$

Note that the dimension of the state to be estimated is $n_u T_{\mathrm{p}} + n_x$.

[1] If u is specified exactly it is possible to derive a reduced order filter which only updates the effect of w and v on the output.

C. Application to model predictive control

The combination of the Kalman filter and the prediction characterisation in (8) are easily combined into a predictive control scheme.

From the last two rows of (8) the future output predictor, $\hat{y}_{\mathrm{f}}$, is easily derived as,

$$\hat{y}_{\mathrm{f}}(k) = E_{x_u} \hat{x}_u + E_{x_y} \hat{x}_y + E_{u\mathrm{f}} u_{\mathrm{f}},$$

where

$$E_{x_u} = (S_{yu} - L_{y\mathrm{f}} L_{u\mathrm{f}}^{-1} S_{uu}),$$
$$E_{x_y} = (S_{yy} - L_{y\mathrm{f}} L_{u\mathrm{f}}^{-1} S_{uy}),$$
$$E_{u\mathrm{f}} = L_{y\mathrm{f}} L_{u\mathrm{f}}^{-1}.$$

To illustrate the approach we will use a simple but relatively common quadratic control cost with output and input costs defined by the matrices $P \in \mathbb{S}_{++}$ and $R \in \mathbb{S}_{+}$ respectively. Input and output constraints, over the length-T_{f} horizon, are specified by the sets $\mathbb{U}$ and $\mathbb{Y}$. Algorithm 1 illustrates the predictive control application of the Kalman filter.

Algorithm 1 Predictive control: SMM-based Kalman filter

Input: $y_{\mathrm{meas}}(k)$
Output: $u(k)$
for every time step, $k = 1, \ldots$ **do**
Kalman filter measurement update:

$$\hat{x}_{uy}(k) = (I - K_{\mathrm{kf}} C_{\mathrm{p}}) \hat{x}_{uy}(k) + K_{\mathrm{kf}} y_{\mathrm{p,meas}}(k)$$

Predictive control step: Solve:

$$\underset{u_{\mathrm{f}}, y_{\mathrm{f}}}{\text{minimise}} \quad \|y_{\mathrm{f}}\|_P + \|u_{\mathrm{f}}\|_R,$$

$$\text{subject to:} \quad y_{\mathrm{f}} = \begin{bmatrix} E_{x_u} & E_{x_y} \end{bmatrix} \hat{x}_{uy}(k) + E_{u\mathrm{f}} u_{\mathrm{f}}, \quad u_{\mathrm{f}} \in \mathbb{U}, \quad y_{\mathrm{f}} \in \mathbb{Y}$$

$$u(k) = \begin{bmatrix} I_{n_u} & 0 & \cdots & 0 \end{bmatrix} u_{\mathrm{f}}.$$

Kalman filter state update:

$$\hat{x}_{uy}(k+1) = A_{\mathrm{p}} \hat{x}_{uy}(k) + B_{u\mathrm{p}} u(k)$$

end for

IV. Simulation Example

To illustrate the application of the data-driven Kalman filter we will apply it to a prediction problem for the longitudinal dynamics of the Boeing 747 aircraft[2]. This example is something of a benchmark in the MPC literature and has previously been used in [7] for comparing data-driven predictive control methods to more standard system identification and model predictive control approaches.

A simplified model of the longitudinal dynamics are given by a 4-state model, with two inputs, throttle, u_1, and elevator angle, u_2 [deg]. The outputs are the longitudinal velocity, y_1 [ft/s], and climb rate, y_2 [ft/s]. The operating conditions

[2] MATLAB code running this comparison is publicly available at: https://doi.org/10.3929/ethz-b-000693730. Simulations of N4SID-Kalman and GDPC use code provided by P. Verheijen.

are an altitude of 40,000 ft. and a horizontal velocity of $V = 774$ ft/s (Mach 0.8). Under these conditions the aircraft has a lightly damped phugoid mode coupling the longitudinal speed and altitude in an oscillatory manner.

The states are defined as the longitudinal velocity, x_1 [ft/s], the downward velocity, x_2 [ft/s], the pitch angular velocity, x_3 [deg/s], and the pitch angle, x_4 [deg]. The continuous-time state-space representation is given by,

$$A = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.74 & 0 \\ 0.02 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.010 & 1 \\ -0.18 & -0.04 \\ -1.16 & 0.598 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 7.74 \end{bmatrix}, \text{ and } D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Although not considered in [7], the effect of turbulence is critical to the safe operation of the aircraft. This is modeled with the Dryden gust model [20] which specifies the horizontal and vertical wind velocity disturbances in terms of their spectral content. As these velocities are also states of the system the disturbance input term is

$$B_w = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_{h,\text{gust}} \\ w_{v,\text{gust}} \end{bmatrix},$$

where the appropriate horizontal and vertical gust spectra are given by the transfer function models,

$$w_{h,\text{gust}} = \sigma_{u,\text{gust}} \sqrt{2L_u/(V\pi)} \frac{1}{1+(L_u/V)s} w_1,$$

$$w_{v,\text{gust}} = \sigma_{v,\text{gust}} \sqrt{2L_v/(V\pi)} \frac{\left(1+\left(2\sqrt{3}L_v/V\right)s\right)}{(1+2(L_v/V)s)^2} w_2.$$

For the simulation we choose the gust intensities corresponding to moderate turbulence at an altitude of 40,000 ft.. The intensities are $\sigma_{u,\text{gust}} = 10$, $\sigma_{v,\text{gust}} = 10$ and the turbulence length scales are $L_u = 1750$ ft. and $L_v = L_u/2$. Figure 1 illustrates typical gust velocities and their effect on the open-loop aircraft dynamics. The phugoid mode is evident in the uncontrolled response. For context note that the allowable operating range for the control study in [7] is $-25 \le y_1 \le 25$ ft/s. and $-15 \le y_2 \le 15$ ft/s.. It's clear that these moderate gusts far exceed the aircraft's desired operational envelope. Control action is required to suppress the gust response in this scenario.

For the prediction and control problem we apply a zero-order hold equivalence transformation using a sample period of 0.1 s. The SMM is created from a noisy experiment of length $K = 2500$ and we choose $T_\text{p} = 30$ and $T_\text{f} = 20$ for the

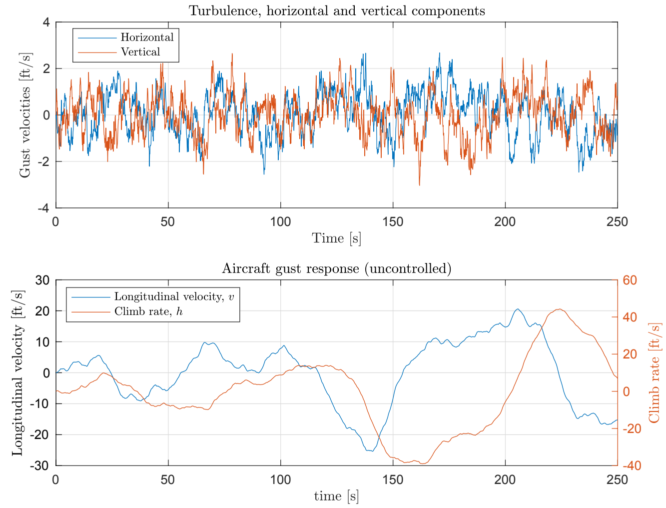

Fig. 1. Representative gust velocities and resulting aircraft velocities.

SMM and Kalman filter implementations. The measurement and process noises have covariances,

$$\Sigma_v = 0.25^2 I_2 \text{ and } \Sigma_w = I_2.$$

Note that this choice of Σ_w reproduces the necessary Dryden gust spectra for the wind disturbance velocities.

To assess the Kalman filter we will consider the gust disturbance scenario in Figure 1. In addition to the gust disturbances, a reference step change (to $y = [10\ 0]^T$) is commanded at $t = 3.0$ seconds. A random input is used for the first 3 seconds. A Monte Carlo study, using 30 simulation runs, is used to compare four control schemes:

- **SMMPC**. The signal matrix model predictive controller described in [17]. In this noise scenario it is equivalent to SPC [8].
- **GDPC**. A regularised data-driven predictive control scheme (GDPC [6]).
- **N4SID-Kal**. Subspace identification (N4SID) using the nominal state dimension ($n_x = 4$) and a Kalman filter with MPC.
- **SMM-Kal**. The SMM-based Kalman filter with MPC given in Algorithm 1.

For each simulation noisy measurement data is used to create a new SMM model. The theoretical results given in [17] are no longer valid, but the comparison is more representative of the practical application. For the SMM-Kalman method we assume that $w(k)$ is known at the modeling stage. The closed-loop simulations have both output measurement noise and unmeasured process noise.

Figures 2 and 3 show the mean and standard deviation of the output response and corresponding actuation trajectories. Figure 4 gives box plots of a several performance indices.

The data-driven methods, SMMPC and GDPC, perform similarly and poorly. This is not surprising as neither contain any characterisation of the effect of the gust disturbance. The past output, y_p, depends on both u_p and the past disturbance, w_p. Attempting to match y_p using only u_p leads to

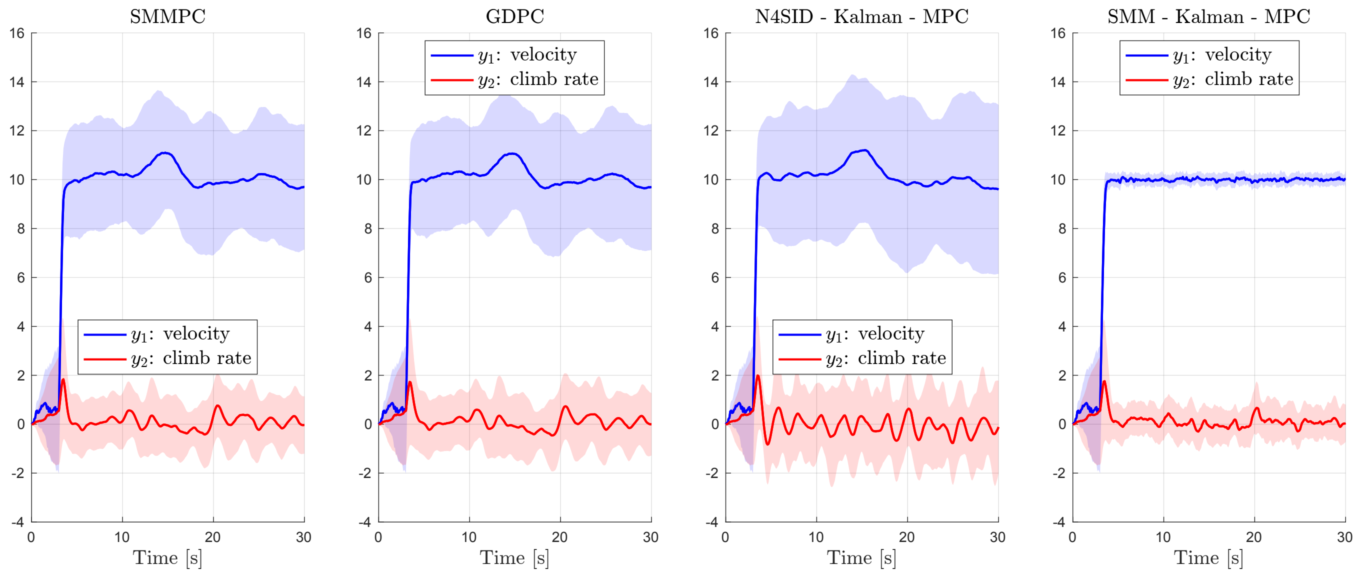

Fig. 2. Step and gust disturbance response. Output trajectories, mean response (solid), ±1 standard deviation (shaded)

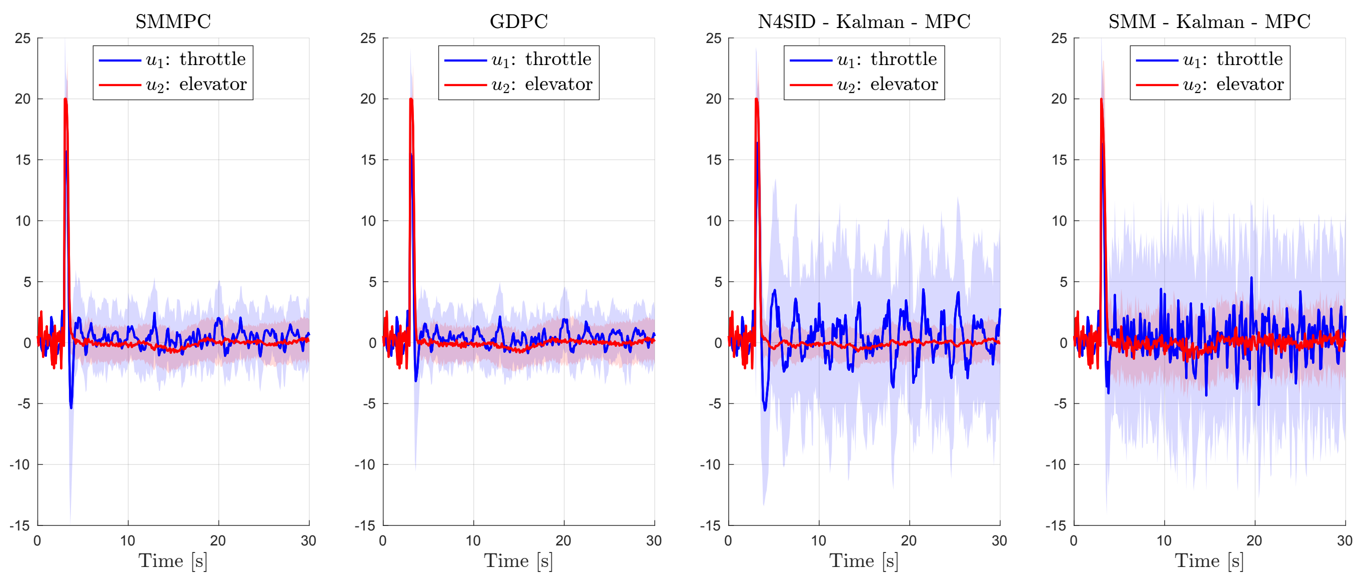

Fig. 3. Step and gust disturbance response. Input trajectories, mean response (solid), ±1 standard deviation (shaded

errors in the predicted future output. The degree of freedom introduced by regularisation in GDPC does not appear to be able to compensate for this. The more classical N4SID identification and Kalman filter method should, in principle, be able to compensate for process disturbances. However it appears that the model identified by N4SID in these circumstances is not sufficiently accurate to allow this. The SMM-based Kalman filter is effectively able to compensate for the unmeasured process noise and performs well.

The actuation trajectories in Figure 3 show that SMMPC and GDPC controllers are essentially not responding to the disturbance. The N4SID based controller is responding to the disturbance but the resulting controller performs poorly. The SMM-based Kalman filter uses additional input energy to effectively control the disturbed system.

V. Conclusions

The inclusion of a Kalman filter in the data-driven predictive control framework makes it easier to apply to a wide range of control formulations, particularly when unmeasured process noise also drives the dynamics. When the measurement and process noise covariance matrices can be estimated the Kalman filter optimally trades off between the two. This feature has largely been absent from prior data-driven control formulations.

Because the Kalman filter, by its recursive nature, averages over all of the past data, it is not necessary to choose a large value of T_{p} in order to achieve adequate noise averaging in the implementation. Smaller values of T_{p} reduce the degree of persistency of excitation needed in defining the SMM matrices. This is advantageous if the user is forced to rely

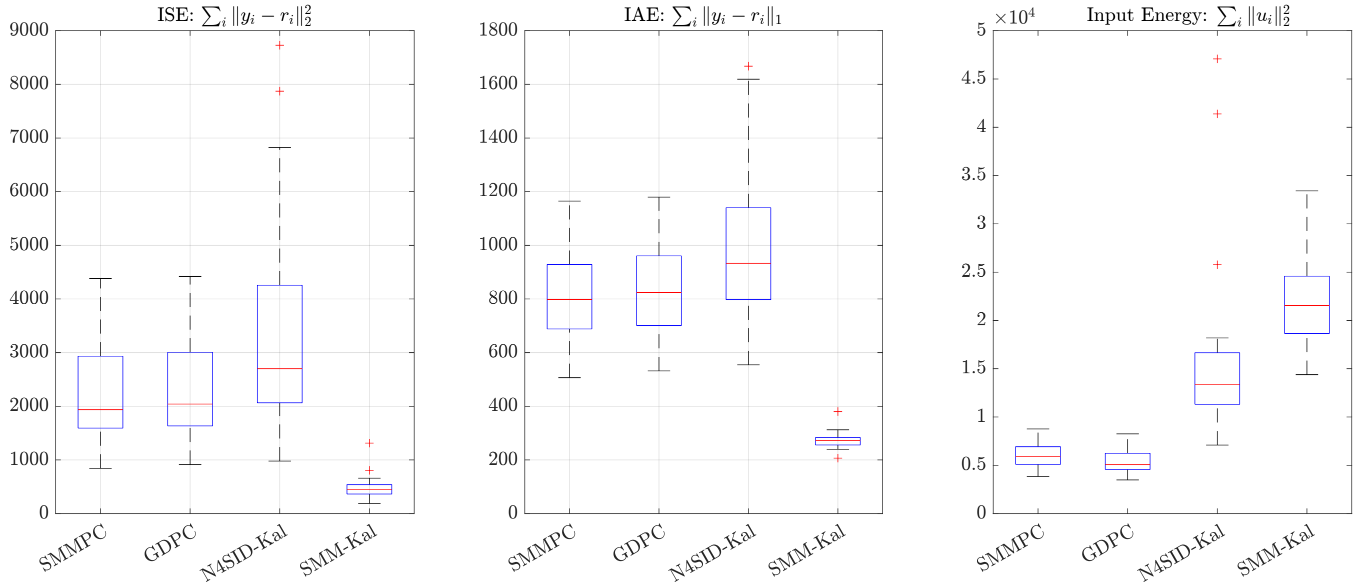

Fig. 4. Performance indices boxplots (Integral square error, Integral absolute error, Input energy)

on historical data which may have limited persistency of excitation.

Over finite horizons the Kalman filter dynamics interact with the control dynamics generating potentially significant transients that are not accounted for when the MPC design is performed independently of the Kalman filter design. The fact that the Kalman filter and MPC problems are strongly interdependent, and that they are both state based, makes it difficult to design them independently. This issue is avoided in the SMM-based Kalman filter formulation given here as the Kalman filter design is carried out with the objective of minimising the prediction errors in the future input-output mapping. This objective is well suited to control over a future horizon.

The quality of the estimates and the subsequent control depend on the input-output trajectory used to define the SMMs. Noise in this trajectory will make the model high order and reduce its predictive accuracy. Good characterisation experiment design is required to give good data-driven control results. This suggests that the experiment design considerations used in system identification are equally applicable to the data-driven control application.

References

[1] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[2] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality and robustness," *IEEE Trans. Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[3] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. European Control Conference*, November 2019, pp. 307–312.

[4] V. Breschi, M. Fabris, S. Formentin, and A. Chiuso, "Uncertainty-aware data-driven predictive control in a stochastic setting," in *IFAC PapersOnLine*, vol. 56-2, 2023, pp. 10 083–10 088.

[5] V. Breschi, A. Chiuso, and S. Formentin, "Data-driven predictive control in a stochastic setting: a unified framework," *Automatica*, vol. 152, p. 110961, 2023.

[6] M. Lazar and P. Verheijen, "Generalized data-driven predictive control: Merging subspace and Hankel predictors," *Mathematics*, vol. 11, p. 2216, 2023.

[7] P. Verheijen, V. Breschi, and M. Lazar, "Handbook of linear data-driven predictive control: Theory, implementation and design," *Annual Reviews in Control*, vol. 56, p. 100914, 2023.

[8] W. Favoreel, B. De Moor, and M. Gevers, "SPC: Subspace predictive control," in *Proc. IFAC World Congress*, 1999, pp. 4004–4009.

[9] M. Yin, A. Iannelli, and R. S. Smith, "Maximum likelihood estimation in data-driven modeling and control," *IEEE Trans. Automatic Control*, vol. 68, no. 1, pp. 317–328, 2023.

[10] ——, "Data-driven prediction with stochastic data: Confidence regions and minimum mean-squared error estimates," in *Proc. European Control Conference*, 2022.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME J. Basic Engineering*, vol. 82 (series D), pp. 35–45, 1960.

[12] A. Tsiamis, N. Matni, and G. J. Pappas, "Sample complexity of Kalman filtering for unknown systems," in *Proc. Machine Learning Research (Conf. Learning for Dynamics and Control)*, vol. 120, 2020.

[13] M. Yin and R. S. Smith, "On low-rank Hankel matrix denoising," in *IFAC-PapersOnLine (System Identification Symposium)*, vol. 54, no. 7, 2021, pp. 198–203.

[14] D. Alpago, F. Dörfler, and J. Lygeros, "An extended Kalman filter for data-enabled predictive control," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 994–999, 2020.

[15] P. Duan, T. Liu, Y. Xing, and K. H. Johansson, "Data-driven Kalman filter using maximum likelihood optimization," arXiv, Dec. 2023.

[16] M. Yin, A. Iannelli, and R. S. Smith, "Stochastic data-driven predictive control: Regularization, estimation, and constraint tightening," arXiv, Dec. 2023.

[17] R. S. Smith, M. Abdalmoaty, and M. Yin, "Optimal Data-Driven Prediction and Predictive Control using Signal Matrix Models," in *Conf. Decision & Control*, Dec. 2024.

[18] M. Yin, A. Iannelli, and R. S. Smith, "Stochastic data-driven predictive control: Regularization, estimation, and constraint tightening," in *Proc. IFAC Symp. System Identification*, Jul. 2024.

[19] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Trans. Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.

[20] H. Liepmann, "On the application of statistical concepts to the buffeting problem," *J. Aeronautical Sciences*, vol. 19, no. 12, pp. 793–800, 1952.