Assimilation of Data for Dynamic Digital Twins by Learning Covariance Information

T. Çağlar[1], I. Altıntaş[1], and R.A. de Callafon[2]

[1] San Diego Supercomputer Center
Universiy of California, San Diego
{tcaglar,altintas}@sdsc.edu
[2] Dept. of Mechanical and Aerospace Enrgineering
Universiy of California, San Diego
callafon@ucsd.edu

Abstract. When computations of the dynamic behavior of a digital twin includes the recursion of an internal state, data assimilation can be used to adjust the numerical values of the state. The optimal linear adjustment of this state on the basis of observations and simulations is known as a Kalman filter, in which an optimal linear gain is computed based on covariance information to minimize the variance on the state error. This paper illustrates that such covariance information can be learned and used to find an optimal trade-off between the observations and simulations for state adjustment. Although the concept of learning covariance information is well understood by the Ensemble Kalman Filter (EnKF), this paper emphasizes the underlying approach how to learn covariance information with the purpose of convergence and minimal variance of the state error. The concept is illustrated for a dynamic digital twins of a linear oscillatory mechanical system and a non-linear dynamic wildfire progression. The examples illustrate that the results on data assimilation heavily depends on the quality of the covariance information.

Keywords: Data Assimilation · Ensemble Kalman Filter · Digital Twin.

1 Introduction

Combining observations from sensors with a digital dynamic simulation model enables the possibility to improve the quality of the data produced by a digital twin as a dynamic replica of a physical process. Although the original concept of a digital twin has been acknowledged almost a decade ago in manufacturing [8–10], the perceived benefits and effectiveness in monitoring, simulation, forecasting and optimization often require more detailed analysis [11]. Recognizing when computations in a digital twin are used to replicate dynamic and time-dependent behavior, such a dynamic digital twin can be combined with data assimilation techniques with very recent applications in power and mechanical systems modeling [2, 4]. In most of these applications it has been recognized that the idea of data assimilation can be formalized with the concept of a Kalman

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

filter or an Ensemble Kalman filter [5], traditionally used in atmospheric data assimilation [14] or wildfire data assimilation [3, 12, 13] applications.

The connection between a (dynamic) digital twin, data assimilation and Kalman filtering has been recognized in the literature [6]. Especially when the dynamic behavior of a digital twin includes the recursion of an internal state, data assimilation with a Kalman filter can be used to adjust the numerical values of the state. Unfortunately, most of the Kalman filter applications involving a digital twin are formulated as conditional probability optimization problems [6]. Although theoretically correct, application to a dynamic digital twin in which states are updated at discrete-time instances becomes not immediately clear. Furthermore, it is insightful to connect classical and explicit solutions of the Kalman gain used in the Kalman filter [1] to dynamic digital twins that have a simple linear dynamic behavior. In addition, it is worthwhile to provide insight on how the Kalman gain can actually be *learned* or *estimated* from the actual data obtained from sensor observations and discrete-time simulations provided by the dynamic digital twin.

This paper gives a short review on how the Kalman gain for the Kalman filter is computed for a dynamic digital twin with a Linear Time Invariant (LTI) dynamic behavior. Subsequently, it is shown how the Kalman gain can be computed using the combination of two covariance matrices that can be learned from the variance of the output and the state produced by a digital twin (forward) simulation. It is also indicated how the covariance matrices can be learned by performing multiple simulations, called ensembles, proving a direct connection to the ensemble Kalman filter. The practical application, and computational requirements of combining data assimilation with a digital twin by either computing or learning covariance information is illustrated for two examples. The first example is an oscillatory linear mechanical system for which reliable position and velocity information must be obtained. The second example involves a dynamic digital twin that can simulate wildfire behavior for which covariance information is learned via ensembles. The examples illustrate that a significant improvement in the quality of the data assimilation of the dynamic digital twin can be achieved, provided accurate and reliable covariance information can be estimated from the data.

2 Problem Statement

For formulating the concept of discrete-time state reconstruction, it is assumed that a recursive progression of a state x_k as function of the discrete-time index k is given by a known Non-Linear (NL) and possibly time varying discrete-time dynamic system

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k) + w_k, \quad x_0 = x(0) \\ y_k &= h_k(x_k) + v_k \end{aligned} \tag{1}$$

where $u_k \in R^m$ is a *known* input, $w_k \in R^m$ is an *unknown* state noise, $y_k \in R^p$ is a *measured* output, $v_k \in R^p$ is an *unknown* measurement noise and where the initial condition x_0 may be assumed to be *unknown*. Furthermore, for notational

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

simplicity it is assumed that the dimension n of $x_k \in R^n$ is known and $x_{k+1} \in R^n$ has the same dimension to allow for recursion of the state x_k through (1).

The unknown initial condition $x_0 \in R^n$ and the notion of noise w_k on the state equation indicates that progression of the state from x_k to x_{k+1} is subjected to uncertainty. The measurements y_k are not perfect either, as they are subjected to measurement noise v_k. Without loss of generality, we may assume a first order moment zero-mean $\mathrm{E}\{w_k\} = 0$, $\mathrm{E}\{v_k\} = 0$ that are uncorrelated between time instance k and $k+1$ (e.g. white noise) with a possible non-stationary second moments modeled by the cross-covariance

$$\mathrm{E}\left\{ \begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} = \begin{bmatrix} W_k & S_k \\ S_k^T & V_k \end{bmatrix} \quad \text{or} \quad \begin{array}{c} \mathrm{E}\{w_k w_k^T\} = W_k \\ \mathrm{E}\{v_k v_k^T\} = V_k \\ \mathrm{E}\{w_k v_k^T\} = S_k \end{array} \tag{2}$$

with time-varying auto-covariances W_k, V_k and cross-covariance matrix S_k.

For data assimilation, the objective is obtain an *optimal* estimate $\hat{x}_{k+1}$ of the actual state x_{k+1} using the observation y_k and the digital twin (1) driven by the input u_k and a (previous) state estimate $\hat{x}_k$. Optimality of the estimate $\hat{x}_{k+1}$ is defined via by ensuring that the stochastic (possibly non-stationary) **state estimation error**[3] $e_{k+1} = x_{k+1} - \hat{x}_{k+1}$ or equivalently

$$e_k = x_k - \hat{x}_k \tag{3}$$

for any value of k, satisfies two important properties:

- The first property is (global) convergence of the (possibly non-stationary) state estimation error
$$\lim_{k \to \infty} \mathrm{E}\{e_k\} = 0$$
as the time index k progresses. It should be mentioned that an estimate $\hat{x}_0$ of the initial condition x_0 can be used to possibly shorten the time of convergence to the value of k_ε where
$$k_\varepsilon = \min_k \mathrm{E}\{e_k\} \leq \varepsilon$$
instead of setting $\hat{x}_0$ simply to $\hat{x}_0 = 0$.
- The second property is (global) optimality of the state estimation error by requiring a minimization of (the trace of) the error covariance matrix
$$P_k = \mathrm{E}\{e_k e_k^T\}$$
for each value of k.

[3] Also often defined as the **adjusted state error** or **Kalman state error** as this pertains to the error on a reconstructed state $\hat{x}_k$ created by either a state estimation, state adjustment or Kalman filter.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

It can be seen that with a desired mean value $\mathrm{E}\{e_k\} = 0$ as $k \to \infty$, only the second moment (e.g. variance) on the state estimation error e_k is used to define the state estimation to be optimal. A filter that minimizes the variance of the state estimation error e_k is also known as a **Kalman filter**. Hence, an optimal state estimation is not necessarily the fastest and visa-versa, but (global) convergence of the state estimation error is required to allow optimality to be defined only by the second moment of the state estimation error.

3 LTI discrete-time system with stationary noise

3.1 Luenberger Observer

To start the explanation of learning covariance information, it is worthwhile to first analyze the most simple case of a dynamic digital twin: a Linear Time-Invariant (LTI) discrete-time system with stationary state and observation noise. For this simple case, an elegant solution to the optimal state estimation problem is given by a Luenberger observer with an optimal (Kalman) observer gain. To explain this elegant solution, consider a LTI discrete-time system given by[4]

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \quad x_0 = x(0) \\ y_k &= Cx_k + v_k \end{aligned} \tag{4}$$

where $f_k(x_k, u_k) = Ax_k + Bu_k + w_k$ and $h_k(x_k, v_k) = Cx_k + v_k$ compared to (1). Stationary white noises v_k and w_k in (4) are represented by

$$\mathrm{E}\left\{ \begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} = \begin{bmatrix} W & S \\ S^T & V \end{bmatrix} \tag{5}$$

with fixed noise auto-covariance matrices W, V and cross-covariance matrix S. With the LTI discrete-time dynamics (4) and covariance information (5), the optimal estimation $\hat{x}_{k+1}$ of the state x_{k+1} is given by a Luenberger observer

$$\begin{cases} \hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k), \quad \hat{x}_0 = \hat{x}(0) \\ \hat{y}_k = C\hat{x}_k \end{cases} \tag{6}$$

where the fixed observer or Luenberger gain L is optimal when set equal to the Kalman gain [1] given by

$$L = (APC^T + S)(CPC^T + V)^{-1} \tag{7}$$

in which the matrix P is the solution to a Discrete Algebraic Ricatti Equation (DARE)

$$P = APA^T - (APC^T + S)(CPC^T + V)^{-1}(APC^T + S)^T + W \tag{8}$$

[4] The analysis is done for a LTI system, but can easily include a non-linear input $b(u_k)$ instead of Bu_k, as either a linear or non-linear input contribution can be accounted for in a Luenberger observer.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

that represents the fixed error covariance matrix

$$P = \mathrm{E}\{e_{k+1}e_{k+1}^T\} = \mathrm{E}\{e_k e_k^T\} \tag{9}$$

of the state estimation error, for which the trace is minimized by the choice of the Kalman gain L.

The derivation of the result in (7) is done by first computing P in (9) as a function of L, using the Luenberger observer given in (6). This leads to a discrete-time Lyapunov equation for which completing the squares allows the selection of a matrix gain L that minimizes the trace of P and leads to the Kalman gain L in (7). Substitution of L in (7) back into the discrete-time Lyapunov equation leads to the DARE in (8) to compute the optimal error covariance matrix P.

3.2 State estimation as a discrete-time filter

The Luenberger observer can also be written as

$$\hat{x}_{k+1} = (A - LC)\hat{x}_k + Bu_k + Ly_k, \quad \hat{x}_0 = \hat{x}(0) \tag{10}$$

clearly showing a discrete-time filter that uses the matrices A, B, C of the discrete-time LTI system in (4), the optimal observer (Kalman) gain L in (7) and both u_k and y_k as inputs. The filter interpretation in (10) is often adopted for the actual implementation of a LTI discrete-time Kalman filter.

Furthermore, the LTI discrete-time dynamics of the state estimation error e_k in (3) is described by

$$\hat{e}_{k+1} = (A - LC)\hat{e}_k + w_k - Lv_k, \quad \hat{e}_0 = x(0) - \hat{x}(0) \tag{11}$$

clearly showing the need for all eigenvalues $|\lambda_i(A - LC)| < 1$ to ensure

$$\lim_{k\to\infty} \mathrm{E}\{e_k\} = 0 \tag{12}$$

It is worth mentioning that if the pair (A, C) is observable, a so-called (stabilizing) solution P to (8) can be computed for which all eigenvalues $|\lambda_i(A-LC)| < 1$ with L given in (7). Clearly, observability of a LTI discrete-time system is a basic property needed to be able to perform state estimation that ensures (12). As a final remark, it is worth recognizing that the Kalman gain L in (7) is chosen as a trade-off to minimize the covariance $P = \mathrm{E}\{e_k e_k^T\}$, but does not necessarily places all eigenvalues of $\lambda_i(A - LC)$ at 0 to ensure the fastest discrete-time convergence of the state estimation error e_k. Only if the covariance matrix $V = \mathrm{E}\{v_k v_k^T\}$ of the output noise v_k is close to zero, a large value of L may be chosen to strongly rely on the output measurements y_k for fast convergence.

3.3 State estimation as a forward model with state adjustment

Yet another way of writing the Luenberger observer is using the concept of a **forward model** given by

$$\begin{cases} \hat{z}_{k+1} = A\hat{x}_k + Bu_k \\ \hat{y}_k = C\hat{x}_k \end{cases} \tag{13}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

that simply creates a "open-loop" or "forward" simulated state $\hat{z}_{k+1}$ and the simulated output $\hat{y}_k$, similar to (6), as a function of a previously estimated state $\hat{x}_k$ and input u_k. With the concept of the forward model in (13), the Luenberger observer in (6) can be written as a recursive procedure

$$\begin{aligned} \textbf{forward model}: & \begin{cases} \hat{z}_{k+1} = A\hat{x}_k + Bu_k, & \hat{x}_0 = \hat{x}(0) \\ \hat{y}_k = C\hat{x}_k \end{cases} \\ \textbf{state adjustment}: & \left\{ \hat{x}_{k+1} = \hat{z}_{k+1} + L(y_k - \hat{y}_k) \right. \end{aligned} \tag{14}$$

that shows the adjustment of the open-loop simulated or **forward state** $\hat{z}_{k+1}$ to an **adjusted state** $\hat{x}_{k+1}$. The adjusted state $\hat{x}_{k+1}$ is equivalent to the state estimate $\hat{x}_{k+1}$ in (6) in which the **Kalman gain** L in (7) and the **output error** $y_k - \hat{y}_k$ between the measurement y_k and the simulated output $\hat{y}_k$ is used. Since the covariance P of the state estimation error e_k remains the same for a LTI system with stationary noise, there is no need for an update of the covariance P over time. The two-step procedure in (14) is often adopted if the forward model is a NL discrete-time system similar to (1), not allowing the state estimation to be written as a single LTI discrete-time filter operation as in (10). In case of an LTI system with unknown time-varying noise covariance matrices as in (2), recursive update of covariance matrices may also be needed.

3.4 Kalman gain in terms of covariance matrices

The computation of the Kalman gain L in (7) requires the explicit computation of the symmetric and positive definite matrix P to the DARE in (8). Knowing that $P = \mathrm{E}\{e_k e_k^T\}$, where e_k is defined as the state estimation error e_k in (3), one may wonder if the Kalman gain can be interpreted in terms of auto- and cross-covariance matrices of error signals. This interpretation would unleash the possibility to compute the Kalman gain directly from covariance information that can be learned from data. To elaborate on this interpretation of the Kalman gain in light of the two-step procedure given in (14), the following error signals are defined.

Following (14), the first error signal is simply the **output error**

$$e_{y,k} = y_k - \hat{y}_k \tag{15}$$

as seen in the Luenberger observer (6) or in the state adjustment step of (14). The second error signal is the **forward state error** or **unadjusted state error** defined by

$$e_{z,k+1} = x_{k+1} - \hat{z}_{k+1} \tag{16}$$

that is *different* from the state estimation error e_{k+1} as defined previously in (3). It should be noted that the forward simulated state $\hat{z}_{k+1}$ is only given by the forward simulation $\hat{z}_{k+1} = A\hat{x}_k + Bu_k$ as part of the forward model in (14), so the forward state error or unadjusted state error $e_{z,k+1}$ in (16) has not been adjusted with any measurements or Kalman gain L.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

With the explicit definition of the above mentioned error signals, the Kalman gain can be interpreted as in terms of auto- and cross-covariance matrices of error signals. With the **output error** signal $e_{y,k}$ in (15) it is clear that

$$\begin{aligned} e_{y,k} &= Cx_k + v_k - C\hat{x}_k \\ &= Ce_k + v_k \end{aligned}$$

so that an auto-correlation of $e_{y,k}$ leads to

$$R_{e_y e_y} = \mathrm{E}\{e_{y,k} e_{y,k}^T\} = CPC^T + V \tag{17}$$

that is part of the definition of the Kalman gain L in (7). With the definition of the **forward state error** signal $e_{z,k+1}$ in (16) it is clear that

$$\begin{aligned} e_{z,k+1} &= Ax_k + Bu_k + w_k - A\hat{x}_k - Bu_k \\ &= Ae_k + w_k \end{aligned}$$

so that a cross-correlation between the **forward state error** $e_{z,k+1}$ and the **output error** $e_{y,k}$ leads to

$$R_{e_z e_y} = \mathrm{E}\{e_{z,k+1} e_{y,k}^T\} = APC^T + S \tag{18}$$

that is also part of the definition of the Kalman gain L in (7). As a result, we can rewrite (7) as

$$L = R_{e_z e_y} R_{e_y e_y}^{-1}, \quad \text{with} \quad R_{e_z e_y} = \mathrm{E}\{e_{z,k+1} e_{y,k}^T\} \quad \text{and} \quad R_{e_y e_y} = \mathrm{E}\{e_{y,k} e_{y,k}^T\} \tag{19}$$

and $e_{y,k}$ indicating the defined **output error** in (15) and $e_{z,k+1}$ indicating the defined **forward state error** or **unadjusted state error** $e_{z,k+1}$ in (16).

The above result shows that the Kalman gain can be interpreted as the product of the *cross-correlation matrix $R_{e_z e_y}$ of the forward state error $e_{z,k+1}$ in* (16) *at time $k+1$ and the output error $e_{y,k}$ in* (15) *at time k* with the *auto-correlation matrix $R_{e_y e_y}$ of the same output error $e_{y,k}$ in* (15) *at time k*. This interpretation is useful when either the covariance matrix P in (8) cannot be computed and/or the linear discrete-time dynamics characterized by the matrices A, B and C is not known. When the discrete-time dynamics is non-linear, a Kalman gain L to adjust the open-loop or forward simulated state $\hat{z}_{k+1}$ to the adjusted state $\hat{x}_{k+1}$ can be computed by estimating the auto- and cross-correlations based on (a finite number of) ensembles of the error signals $e_{y,k}$ and $e_{z,k+1}$. This observation is the basis of the ensemble Kalman Filter (EnKF) and the principle of learning covarinace information via ensembles.

4 Ensemble Kalman Filter

Let us go back to the original problem formulation of discrete-time state reconstruction for the NL discrete-time system in (1). The interpretation of the Luenberger observer as a two-step procedure in (14) and the representation of

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

the Kalman gain as a product of an cross- and an auto-covariance matrix in (19) allows us to formulate state reconstruction for (1). Although this is a viable extension of the Luenberger observer with a Luenberger gain L equivalent to the (linear) Kalman gain, information on the cross- and an auto-covariance matrices now has to be estimated from ensembles of the forward state error $e_{z,k+1}$ at time $k+1$ in (16) and the output error $e_{y,k}$ at time k in (15).

Ensembles are needed, as no explicit computations of the cross- and an auto-covariance matrices are possible due to the non-linearity of (1) and replicated in a forward model. Ensembles to create an estimate $\bar{R}_{e_z e_y}$ for $R_{e_z e_y} = \mathrm{E}\{e_{z,k+1} e_{y,k}^T\}$ and an estimate $\bar{R}_{e_y e_y}$ for $R_{e_y e_y} = \mathrm{E}\{e_{y,k} e_{y,k}^T\}$ can be done by making the following two basic assumptions on the probability distribution of the state estimate $\hat{x}_k$, the state noise w_k and the output noise v_k:

1. Assume the given (previous) state $\hat{x}_k$ at time k has an uncertainty characterized by a mean value $\bar{x}_k$ and a variance X_k according to a Normal probability distribution

$$\hat{x}_k \sim \mathcal{N}(\bar{x}_k, X_k), \quad \bar{x}_k = \mathrm{E}\{\hat{x}_k\} \quad \text{and} \quad X_k = \mathrm{E}\{(\hat{x}_k - \bar{x}_k)(\hat{x}_k - \bar{x}_k)^T\}$$

2. Assume the zero mean valued state noise w_k and output noise v_k also have a Normal probability distribution with a non-stationary variance

$$\mathrm{E}\left\{ \begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} = \begin{bmatrix} W_k & S_k \\ S_k^T & V_k \end{bmatrix} \quad \text{or} \quad \begin{matrix} \mathrm{E}\{w_k w_k^T\} = W_k \\ \mathrm{E}\{v_k v_k^T\} = V_k \\ \mathrm{E}\{w_k v_k^T\} = S_k \end{matrix}$$

as given earlier in (2).

Under these assumption, N **ensembles** of $\hat{x}_k^j$, v_k^j and w_k^j for $j = 1, 2, \ldots, N$ at time instance k can be created by taking samples from the above probability distributions. Based on these ensembles, the Ensemble Kalman filter (EnKF) can be formulated again as a recursive Luenberger observer similar to (14). However, a Kalman gain $\bar{L}$ is now computed as a product of *estimated* cross-covariance matrix $\bar{R}_{e_z e_y}$ and the inverse of an *estimated* auto-covariance matrix $\bar{R}_{e_y e_y}$. The state adjustment is done from an *estimated* mean value $\bar{z}_{k+1}$ of the open-loop/forward simulated state ensembles $\hat{z}_{k+1}^j$ and an *estimated* mean value $\bar{y}_k$ of the output ensembles $\hat{y}_k^j$ over $j = 1, 2, \ldots, N$. In summary, the EnKF can be summarized by the recursive steps:

$$\textbf{forward model}: \begin{cases} \hat{z}_{k+1}^j = f_k(\hat{x}_k^j, u_k) \\ \hat{y}_k^j = h_k(\hat{x}_k^j) \end{cases} \quad \text{for} \quad j = 1, 2, \ldots, N$$

$$\textbf{state adjustment}: \begin{cases} \bar{x}_{k+1} = \bar{z}_{k+1} + \bar{L}(y_k - \bar{y}_k) \\ \bar{z}_{k+1} = \frac{1}{N} \sum_{j=1}^{N} \hat{z}_{k+1}^j \\ \bar{y}_k = \frac{1}{N} \sum_{j=1}^{N} \hat{y}_k^j \\ \bar{L} = \bar{R}_{e_z e_y} \bar{R}_{e_y e_y}^{-1} \end{cases} \tag{20}$$

$$\textbf{covariance update}: \begin{cases} X_{k+1} = \frac{1}{N} (\hat{x}_{k+1}^j - \bar{x}_{k+1})(\hat{x}_{k+1}^j - \bar{x}_{k+1})^T \\ \hat{x}_{k+1}^j = \hat{z}_{k+1}^j + \bar{L}(y_k - \hat{y}_k^j) \end{cases} \quad \text{for} \quad j = 1, 2, \ldots, N$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

in which the subsequent computational steps to in the state adjustment are explained in the following.

1. With the N ensembles of the open-loop/forward state $\hat{z}^j_{k+1}$ and output $\hat{y}^j_k$ obtained by the forward model in (20), one can learn or estimate the mean value $\bar{z}_{k+1}$ of the forward state $\hat{z}_{k+1} = \mathrm{E}\{\hat{z}^j_{k+1}\}$ via

$$\bar{z}_{k+1} = \frac{1}{N}\sum_{j=1}^{N} \hat{z}^j_{k+1}$$

In addition, the mean value $\bar{y}_k$ of the output $\hat{y}_k = \mathrm{E}\{\hat{y}^j_k\}$ can be estimated via

$$\bar{y}_k = \frac{1}{N}\sum_{j=1}^{N} \hat{y}^j_k$$

With the estimated mean values, we obtain N ensembles of the forward state error and output error signals

$$\begin{aligned} e^j_{z,k+1} &= \hat{z}^j_{k+1} - \bar{z}_{k+1} + w^j_k \\ e^j_{y,k} &= \hat{y}^j_k - \bar{y}_k + v^j_k \end{aligned} \tag{21}$$

in which also the noise ensembles w^j_k and v^j_k for $j = 1, 2, , \ldots, N$ are included.
2. With the N ensembles of the forward state error $e^j_{z,k+1}$ and the output error $e^j_{y,k}$ for $j = 1, 2, \ldots, N$, an estimate $\bar{R}_{e_z e_y}$ of the cross-covariance matrix $R_{e_z e_y} = \mathrm{E}\{e_{z,k+1} e^T_{y,k}\}$ can be learned via

$$\bar{R}_{e_z e_y} = \frac{1}{N}\sum_{j=1}^{N} e^j_{z,k+1} {e^j_{y,k}}^T \tag{22}$$

and an estimate $\bar{R}_{e_y e_y}$ of the auto-covariance matrix $R_{e_y e_y} = \mathrm{E}\{e_{y,k} e^T_{y,k}\}$ can be learned via

$$\bar{R}_{e_y e_y} = \frac{1}{N}\sum_{j=1}^{N} e^j_{y,k} {e^j_{y,k}}^T \tag{23}$$

As indicated in (20), the computed estimates $\bar{R}_{e_z e_y}$ and $\bar{R}_{e_y e_y}$ of respectively the cross-covarariance $R_{e_z e_y}$ and auto-covariance $R_{e_y e_y}$ can now be used to compute an estimate $\bar{L}$ of the Kalman gain L via $\bar{L} = \bar{R}_{e_z e_y} \bar{R}^{-1}_{e_y e_y}$ to facilitate the state adjustment $\bar{x}_{k+1} = \bar{z}_{k+1} + \bar{L}(y_k - \bar{y}_k)$. It is worth noting that $N \geq p$, where p is the number of outputs or the length of y_k and $\hat{y}_k$, to ensure $\bar{R}_{e_y e_y}$ is invertible. In general, it is recommended that $N \geq 10p$ to provide more accurate auto and cross-covariance estimates. With the state estimation error now defined by $\bar{e}_{k+1} = x_{k+1} - \bar{x}_{k+1}$, it is also worth noting that the resulting Kalman gain $\bar{L}$ in (20) does not necessarily guarantee that $\lim_{k\to\infty} \mathrm{E}\{\bar{e}_k\} = 0$, due to the non-linearity of (1) and replicated in the forward model of (20). So care must be given to relatively large Kalman gain $\bar{L}$ that could possible destabilize or increase the state estimation error $\bar{e}_{k+1}$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

5 Illustration for oscillatory mechanical system

To illustrate the performance of data assimilation by learning covariance information, we first consider a 1 degree of freedom (DOF) single mass/spring/damper mechanical system. The reason for choosing this simple example is due to the clear interpretation of the states of the resulting second order dynamical system: *position* and *velocity*. The dynamic digital twin is a LTI 10Hz sampled Zero Order Hold discrete-time model and for a mass $m = 5$ kg, spring $k = 2.5$ N/m and a damper $d = 0.5$ Ns/m, the LTI system is given by the state-space model

$$\begin{cases} x_{k+1} = Ax_k + B\tilde{u}_k, \quad x_0 = x(0) \\ y_k = Cx_k + D\tilde{u}_k + n_k \end{cases} \quad \text{where} \quad \tilde{u}_k = u_k + m_k \quad \text{and}$$
$$A = \begin{bmatrix} 0.9975 & 0.0994 \\ -0.0497 & 0.9876 \end{bmatrix}, \; B = \begin{bmatrix} 0.0010 \\ 0.0199 \end{bmatrix}, \tag{24}$$
$$C = \begin{bmatrix} 1 & 0 \\ -0.5 & -0.1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix}$$

in which both the *position* and *acceleration* are chosen as observation y_k. The independent noise signal m_k with a covariance Λ_m indicates a noise present on the applied known input u_k and the n_k with a covariance Λ_n is a measurement noise on the observations y_k. This can be written in the standard form of (4) with $w_k = Bm_k$ and $v_k = n_k + Dm_k$ with the following noise covariance matrices

$$\begin{aligned} W &= \mathrm{E}\{w(k)w(k)^T\} = B\Lambda_m B^T \approx \tfrac{1}{1000}\begin{bmatrix} 0.0050 & 0.0990 \\ 0.0990 & 1.9768 \end{bmatrix} \\ V &= \mathrm{E}\{v(k)v(k)^T\} = D\Lambda_m D^T + \Lambda_n \approx \tfrac{1}{1000}\begin{bmatrix} 100 & 0 \\ 0 & 300 \end{bmatrix} \\ S &= \mathrm{E}\{w(k)v(k)^T\} = B\Lambda_m D^T \approx \tfrac{1}{1000}\begin{bmatrix} 0 & 0.9963 \\ 0 & 19.8838 \end{bmatrix} \end{aligned} \tag{25}$$

Step-wise changes in the known force input u_k leads to the measured position and acceleration measurement shown in Figure 1. It can be observed that the measurement are fairly noisy and the objective is to use both input and output measurements to produce an estimate of the position and velocity.

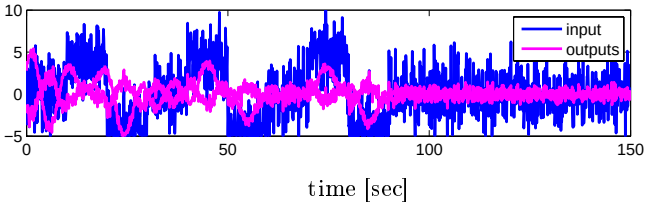

Fig. 1. Noisy force input and position, acceleration output measurements to be used for data assimilation of position and velocity of a 1DOF oscillatory mechanical system.

Data assimilation to obtain the position and velocity can be as simple as taking the position measurement as-is, and either digitally differentiating the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

position measurement or digitally integrating the acceleration measurements. Unfortunately, either choices will be far from optimal, especially if the observations y_k are subjected to noise, as illustrated in Figure 2 as a reference.

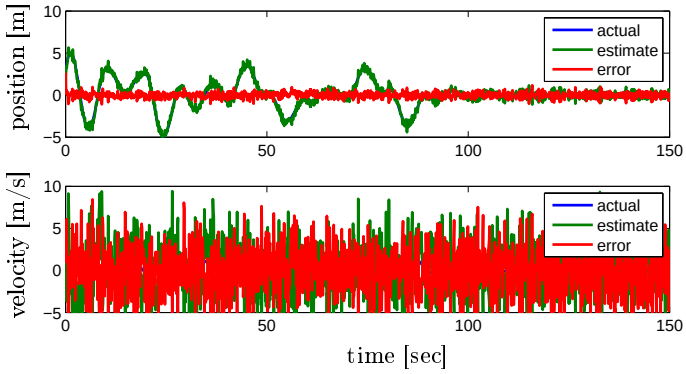

Fig. 2. Noise performance of data assimilation for a 1DOF mechanical system to estimate velocity based on digital differentiation of position measurements.

Alternatively, the use of the correct Kalman gain L optimizes the procedure of assimilating the position and acceleration data into a position and velocity update. The results are summarized in Figure 3 for comparison with Figure 2.

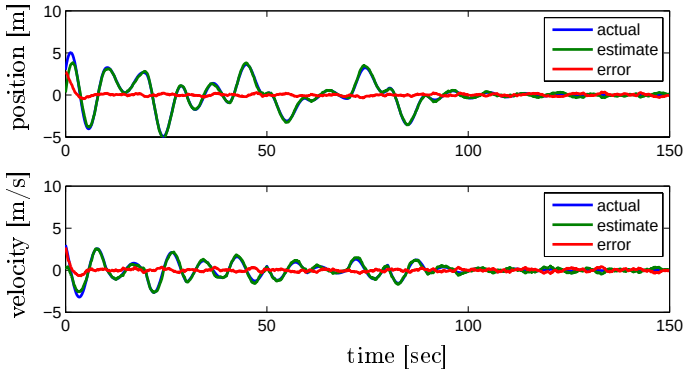

Fig. 3. Noise performance of data assimilation for a 1DOF mechanical system to estimate position and velocity based on noisy input/output measurements by learning of covariance matrices and adjusting the Kalman gain.

With the full information on the state matrices in (24) and the noise covariance matrices in (25) of the linear dynamical digital twin, the optimal Kalman gain L in (7) can be directly computed by solving P for the DARE in (8). Alternatively, the procedure of creating ensembles and learning the auto-covariance information $R_{e_y e_y}$ in (17) and cross-covariance $R_{e_z e_y}$ in (18) can be used to

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

compute the same Kalman gain L in (19). It can be observed from Figure 3 that the state error converges (on average) to the actual position and velocity measurement. Although the convergence is in the order of several seconds, the variance on the position and velocity estimation has been significantly reduced compared to the results shown earlier in Figure 2. This trade-off between state error convergence and reduction of the variance is due to the ratio between the covariance matrices $R_{e_y e_y}$ and $R_{e_z e_y}$ learned from data that eventually determines the Kalman gain L in (19).

6 Illustration on wildfire data

The approach of adjusting the internal state for data assimilation can also be applied to a more complex and non-linear dynamic digital twin that simulates the dynamic growth of a wild fire. The FARSITE wild fire simulation tool [7] can be considered to be a (non-linear) time varying discrete-time dynamic system (1) with a recursive state of real valued eastern- and northern-coordinates of a fire perimeter x_k at time step k. Forward simulation of a fire perimeter x_{k+1} at time step $k+1$ is done by using information on surface fuels, topography and fuel adjustment factors, collectively combined in an environmental parameter dependent function $f_k(\cdot)$ at time step k. Important inputs u_k at time step k include wind speed and wind direction. Dynamically, the FARSITE wild fire simulation tool is a non-linear discrete-time integrator that can be started from an initial fire perimeter x_0.

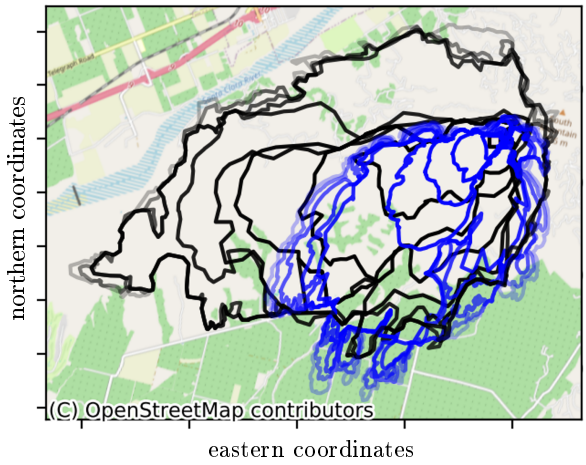

Fig. 4. Unadjusted simulation of the Maria 2019 wild fire (blue lines) compared to the measured fire perimeter observations (black lines).

Subsequent fire perimeters x_k can be computed by the implicit computation of the environmental parameter dependent function $f_k(\cdot)$, along with wind data

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

u_k. As been demonstrated in earlier work, running a digital twin of a wildfire simply open-loop over multiple subsequent time indexes $k+1, k+2, \ldots$ without adjustment based on fire perimeter measurements $y_{k+1,k+2}, \ldots$ may lead to compounding errors in the state estimates $\hat{x}_{k+1}, \hat{x}_{k+2}, \ldots$. Such compounding errors will lead to divergence of the predicted wildfire perimeters $\hat{y}_{k+1}, \hat{y}_{k+2}, \ldots$ as the time index progresses. This is evident from a simulation of the 2019 Maria fire depicted in Figure 4. In Figure 4 a comparison is made between the progression of "true" measured/noisy fire perimeters (black lines) and an unadjusted digital twin simulation of the same fire, starting at a different initial fire perimeter x_0. It can be observed that the unadjusted fire simulations diverge from the measured/noisy observations.

The solution to the problem of divergence is well understood and can be solved by assimilating the observations $y_{k+1}, y_{k+2}, \ldots$ into the prediction of the state estimates $\hat{x}_{k+1}, \hat{x}_{k+2}, \ldots$. In essence, this is done by learning the covariance matrices $R_{e_z e_y}$, $R_{e_y e_y}$ via the estimates given in (22), (23) via N ensembles of the forward state error and the output error in (21) and adjusting the state $\hat{x}_{k+1}$ with the Kalman gain in (19). Again with the correct estimation of the covariance information, the divergence of the wildfire simulation can be significantly reduced as illustrated in Figure 5 for comparison with Figure 4.

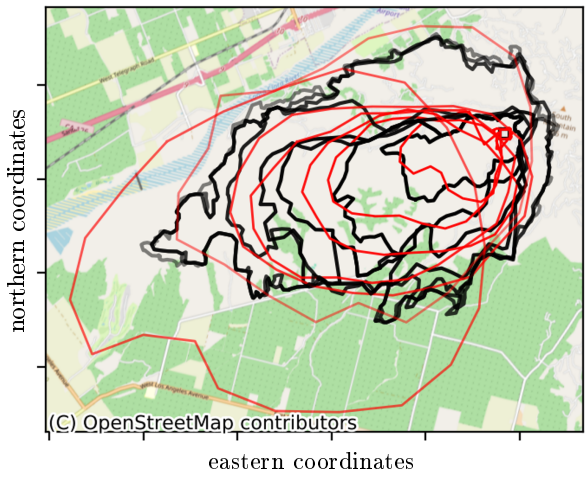

Fig. 5. Hourly sdjusted data assimilation of the Maria 2019 wild fire (red lines) compared to the measured fire perimeter observations (black lines).

A few important recommendations on the learning of the covariance matrices $\bar{R}_{e_z e_y}$ and $R_{e_z e_y}$ in (20) should be highlighted here to ensure the quality of the data assimilation results for a wildfire similar to Figure 5.

– For digital twins of wildfires where the dimension or the orientation of the vertices of the forward simulated ensembles $\hat{y}_k^j$ are *different* from the dimension of the observation y_k, an interpolation and realignment of the observa-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

tion y_k is needed to be able to compute the average $\bar{y}_k$, the cross covariance estimate $\bar{R}_{e_y e_y}$ and the state adjustment in (20).
- For digital twins of wildfires where the dimension or the orientation of the vertices of the forward simulation ensembles $\hat{z}^j_{k+1}$ are *different*, an interpolation and realignment of the ensembles $\hat{z}^j_{k+1}$ is needed to be able to compute the average $\bar{z}_{k+1}$ and the cross covariance estimate $\bar{R}_{e_x e_y}$ in (20).
- Last, but not least, to ensure that the estimate $\bar{R}_{e_y e_y}$ is invertible to compute the Kalman gain $\bar{L}$ in (20), the number of ensembles N must satisfy $N \geq p$, where p is the number of data points in the measured (and possibly interpolated) fire perimeter y_k and forward simulated fire perimeter $\hat{y}_k$. In general, it is recommended that $N \geq 10p$ to provide more accurate auto and cross-covariance estimates, otherwise an incorrect Kalman gain $\bar{L}$ is obtained at each data assimilation step.

The first two observations are important to ensure the Kalman gain $\bar{L}$ and the resulting state adjustment $\bar{x}_{k+1} = \bar{z}_{k+1} + \bar{L}(y_k - \bar{y}_k)$ in (20) are computed correctly. The latter condition imposes severe restriction on the number of points (resolution) p of the fire perimeter in case the *computational resource* or *computational time* for data assimilation is limited. For real-time predictions, reduction of the resolution of the data assimilated wildfire perimeter may be needed. The reduced resolution can also be observed in Figure 5 to allow for computations to be completed within the hourly time frames of each data assimilation step.

7 Conclusions

The quality of the data produced by a dynamic digital twin as a dynamical replica of a physical process can be significantly improved by adjusting the open-loop or forward simulations of the digital twin with observations from the physical process. The idea of adjustment of the simulation can be formalized with the concept of data assimilation, that is founded upon an adjustment of the internal state of the dynamic digital twin using a Kalman gain. For that purpose, the dynamic behavior of a digital twin must includes the recursion of an internal state that can be adjusted using the computed Kalman gain in a Kalman filter.

For linear dynamic digital twins, the Kalman gain can be computed explicitly with information on the linear dynamic behavior and covariance information on the state and observation noise. Without such information, the Kalman gain can be computed as a combination of two covariance matrices that can be learned from the covariance information of the output vector and the state vector produced by the digital twin (forward) simulation. The covariance information can be learned by performing ensemble simulations, each used to estimate first moment (average) and second moment (variance). Ensembles may be computationally demanding and therefore the size of the output vector may have to be limited. Illustration of combining data assimilation with a digital twin of a mechanical system and a wildfire simulation show promising results, indicating the broad application of the approach.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9

Acknowledgments. This work was funded with support from the California Governor's Office of Emergency Services (Cal OES) Fire Integrated Real-time Intelligence System (FIRIS) program. The authors would like to thank the WIFIRE and Cal OES teams for their collaboration, in particular, Daniel Crawl and Jessica Block.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Anderson, B.D.O., Moore, J.B.: Kalman Filtering: Whence, What and Whither?, pp. 41–54. Springer Berlin Heidelberg, Berlin, Heidelberg (1991)
2. Baldassarre, A., Dion, J.L., Peyret, N., Renaud, F.: Digital twin with augmented state extended Kalman filters for forecasting electric power consumption of industrial production systems. Heliyon **10**(6) (2024)
3. Beezley, J., Mandel, J.: An ensemble Kalman-particle predictor-corrector filter for non-Gaussian data assimilation. Lec. Notes in Comp. Science **5545**, 470–478 (2009)
4. Branlard, E., Jonkman, J., Brown, C., Zhang, J.: A digital twin solution for floating offshore wind turbines validated using a full-scale prototype. Wind Energy Science **9**, 1–24 (2024)
5. Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer-Verlag, Berlin (2009)
6. Feng, H., Gomes, C., Larsen, P.G.: Model-based monitoring and state estimation for digital twins: The Kalman filter (2023), https://arxiv.org/abs/2305.00252
7. Finney, M.: FARSITE: Fire area simulator-model development and evaluation. Tech. Rep. RMRS-RP-4 Revised, U.S. Dept. of Agriculture, Forest Service, Rocky Mountain Research Station (2004)
8. Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine **51**(11), 1016–1022 (2018), 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018
9. Liu, M., Fang, S., Dong, H., Xu, C.: Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems **58**, 346–361 (2021)
10. Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.: About the importance of autonomy and digital twins for the future of manufacturing. In: 15th IFAC Symp. on Information Control Problems in Manufacturing. vol. 48, pp. 567–572 (2015)
11. Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., Calinescu, A.: Digital twins: State of the art theory and practice, challenges, and open research questions. Journal of Industrial Information Integration **30**, 100383 (2022)
12. Srivas, T., Artés, T., de Callafon, R., Altintas, I.: Wildfire spread prediction and assimilation for FARSITE using ensemble Kalman filtering. Procedia Computer Science **80**, 897–908 (2016)
13. Tan, L., de Callafon, R.A., Block, J., Crawl, D., Çağlar, T., Altıntaş, I.: Estimation of wildfire wind conditions via perimeter and surface area optimization. Journal of Computational Science **61**, 101633 (2022)
14. Todling, R., Cohn, S.E.: Suboptimal schemes for atmospheric data assimilation based on the Kalman filter. Monthly Weather Review **122**(11), 2530âĂŞ–2557 (1994)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_9