

Thermal state estimation of fused deposition modeling in additive manufacturing processes using Kalman filters

Youngkyu Kim¹ | David Alcantara | Tarek I. Zohdi

Department of Mechanical Engineering,
University of California, Berkeley,
California

Correspondence

Youngkyu Kim, Department of
Mechanical Engineering, University of
California, Berkeley, CA 94720-1740.
Email: youngkyu_kim@berkeley.edu

Funding information

Advanced Robotics for Manufacturing,
Grant/Award Number: ARM 17-01-F-C07

Summary

In this paper, a data assimilation method using a linear Kalman filter is presented to monitor the thermal state of a part during a fused deposition modeling process. The system model is found by discretizing the heat equation with the finite difference method. Infrared (IR) camera measurements are then combined with the system model via a linear Kalman filter. As a result of the data assimilation, the accuracy of monitoring the thermal field is improved, which is important for the qualification of bonding strength between layers. Furthermore, using the linear Kalman filter can allow for closed-loop monitoring system because an IR camera can be guided to a location where the uncertainty of estimation is high. The closed-loop monitoring system will be employed as a feedback system for adaptive control of system parameters during printing processes for the improvement of part quality.

KEYWORDS

finite difference methods, fused deposition modeling process, heat equation, IR camera measurements, Kalman filter

1 | INTRODUCTION

Among polymer additive manufacturing (AM) processes, material extrusion, often also known as fused deposition modeling (FDM) or fused filament fabrication is the most widely used technology. In the FDM process, an object is built by selectively depositing melted polymer material in a predetermined path layer-by-layer which rapidly hardens into rigid material.¹

A well-known issue of the FDM process is the mechanical anisotropy of final parts due to weak bonding between layers. The layer interface is where the failure typically occurs.² The interfacial temperature determines the quality of the bonding between layers. If the temperature is too high, thermal degradation may occur. On the other hand, if the temperature is very low, interlayer bonding or adhesion can be weak. Accordingly, it is a point of interest to monitor the thermal state of parts during the FDM process.^{3,4} In order to avoid weak bonding between layers and thermal degradation, it is critical to monitor the thermal state of the part being printed. Currently, there is no satisfactory technical solution for the FDM process which can provide in situ thermal information.

This research proposes to combine a simulation of the physical process with sensor measurements using a linear Kalman filter. The Kalman filter^{5,6} is an estimation algorithm developed by Kalman and Bucy in 1960s which is based on a linear state-space model and a series of physical measurements. By applying the linear Kalman filter, locally measured temperature data from an IR camera is fused with a thermal simulation. This proposed system will capture more information than a system solely counting on the measurements and improve accuracy of the thermal simulations because

the IR camera is able to observe the unexpected effects that the thermal simulation does not capture, including imprecise knowledge of boundary conditions and variation of material properties. A Kalman filter will also reduce uncertainties both in the temperature measurements and in the thermal model.⁷

Data assimilation using a Kalman filter has been used in various applications such as oceanography and meteorology.⁸ In Reference 9, an interesting application of a Kalman filter as an output-only method is presented, where an unscented Kalman filter developed by Julier et al¹⁰ estimates displacement of structures and also identifies model parameters. In our work, we use a linear Kalman filter as a data assimilation technique for the estimation of a thermal field with input sources such as cooling fan speed and ambient temperature in the Kalman filter algorithm. We apply this data assimilation technique to monitoring of the FDM process. This can be interpreted as an enhanced digital twin solution. The digital twin is a thermal model of the FDM process which is enhanced by updating the state estimate continuously in real-time through sensor measurements. This combined system will output the accurate thermal state estimation and the uncertainty measurements on that estimation, which will enable real-time monitoring of the FDM process. The accurately monitored temperature profile during the printing process will help monitor the quality of printed parts. The uncertainty measure on the thermal state estimation output by the linear Kalman filter can be used to control the position of the IR cameras, guiding them to areas of interest where uncertainty is high.

Due to the lack of a monitoring system for the FDM process, current FDM processes run in an open-loop configuration but the proposed monitoring system will enable closed-loop control of printing parameters such as nozzle temperature and cooling fan speed. By using the information gathered by the monitoring system, printing parameters can be adjusted to maintain an ideal interfacial temperature for strong interlayer bonding without thermal degradation and without large temperature gradients which induce undesirable thermal stresses.

Figure 1 shows the integrated procedure of a suggested solution for monitoring the FDM process using the Kalman filter. To run the Kalman filter algorithm and test performance of the proposed monitoring system without physical experiments, a numerical experiment is presented in this work. The virtual ground truth data and the measurements are generated by adding random noise to the numerical solution of the FDM process. Using the generated measurements, the data fusion process of the thermal field for the FDM process using the linear Kalman filter is performed. The estimated temperature and the virtual ground truth temperature are then compared.

2 | LINEAR KALMAN FILTER

2.1 | State-space model

Two models are used in a linear Kalman filter: the system model and the observation model. The system model is given by a finite difference spatial discretization and forward Euler time integration of the heat equation. This system model is a linear dynamic system which is expressed as

$$\mathbf{X}_{k+1} = \mathbf{A}_k \mathbf{X}_k + \mathbf{F}_k + \mathbf{w}_k. \quad (1)$$

Here single subscript k denotes the time step, $\mathbf{X}_k \in \mathbb{R}^{n_k}$ denotes the state vector, where n_k is the dimension of the state vector at time step k and increases as the FDM process adds material, $\mathbf{A}_k \in \mathbb{R}^{n_k \times n_k}$ denotes the discrete Laplacian operator and is called the state transition matrix which represents how state vectors propagate in discrete time, $\mathbf{F}_k \in \mathbb{R}^{n_k}$ denotes the input vector which originate from the source terms and boundary conditions, and $\mathbf{w}_k \in \mathbb{R}^{n_k}$ denotes the system noise.

Measuring the state vector directly leads to the observation model which is given by

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{v}_k, \quad (2)$$

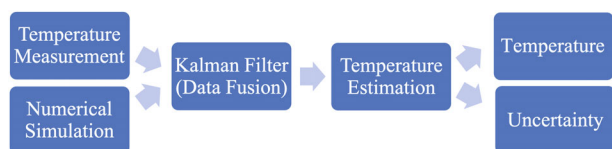


FIGURE 1 Integrated procedure

where $\mathbf{z}_k \in \mathbb{R}^m$ is the observation or measurement vector and the dimension m is set by a user and a sensor specification, and $\mathbf{H}_k \in \mathbb{R}^{m \times n_k}$ is the observation matrix, and $\mathbf{v}_k \in \mathbb{R}^m$ is the measurement noise.

The system noise and measurement noise are assumed to be uncorrelated and zero-mean Gaussian white noise. First, the system noise and the measurement noise are assumed uncorrelated to each other because the measurement noise is originated from a IR camera sensor, whereas the system noise is due to all other effects but for the sensor error. For example, there are model errors, numerical errors, environmental errors, etc., rather than sensor noise. Next, we consider sources of noises as random and independent event at each time, so noises at different times are assumed uncorrelated. Since the randomness in nature is often well described with the Gaussian distribution, we assume that noises follow the Gaussian distribution. Furthermore, the system noise is assumed unbiased because our model which is based on physics is well aligned with the setup of the FDM process. If the sensor is calibrated well, the measurement noise can be assumed unbiased.

2.2 | Formulation

Assuming the system noise \mathbf{w}_k and measurement noise \mathbf{v}_k follow a Gaussian distribution, the linear Kalman filter is a minimum variance estimator of a linear dynamic system.¹¹ With the condition,

$$E(\mathbf{x}_{k,k}) = E(\mathbf{X}_k), \quad (3)$$

which means an estimator is unbiased, a minimum variance unbiased estimator is given by

$$\mathbf{x}_{k,k} = \arg \min_{\mathbf{x}_{k,k}^*} E(\|\mathbf{x}_{k,k}^* - \mathbf{X}_k\|^2 | \mathbf{z}_1, \dots, \mathbf{z}_k) = E(\mathbf{X}_k | \mathbf{z}_1, \dots, \mathbf{z}_k), \quad (4)$$

where $E(\|\mathbf{x}_{k,k} - \mathbf{X}_k\|^2)$ is called the variance of error, and we have the following relationship:

$$E(\|\mathbf{x}_{k,k} - \mathbf{X}_k\|^2) = \text{trace}(E((\mathbf{x}_{k,k} - \mathbf{X}_k)(\mathbf{x}_{k,k} - \mathbf{X}_k)^T)). \quad (5)$$

Given a series of observations, the Kalman filter estimates the state vector by minimizing the variance of error, which is equivalent to the trace of error covariance. The prediction of the state vector, $\mathbf{x}_{k+1,k}$ is defined as an expectation of \mathbf{X}_{k+1} , given observations, $\mathbf{z}_1, \dots, \mathbf{z}_k$. Then, the prediction of the state vector, $\mathbf{x}_{k+1,k}$ is given by

$$\begin{aligned} \mathbf{x}_{k+1,k} &= E(\mathbf{X}_{k+1} | \mathbf{z}_1, \dots, \mathbf{z}_k) \\ &= E(\mathbf{A}_k \mathbf{X}_k + \mathbf{F}_k + \mathbf{w}_k | \mathbf{z}_1, \dots, \mathbf{z}_k) \\ &= \mathbf{A}_k E(\mathbf{X}_k | \mathbf{z}_1, \dots, \mathbf{z}_k) + \mathbf{F}_k + E(\mathbf{w}_k | \mathbf{z}_1, \dots, \mathbf{z}_k) \\ &= \mathbf{A}_k \mathbf{x}_{k,k} + \mathbf{F}_k, \end{aligned} \quad (6)$$

with the assumption of zero-mean Gaussian noise, \mathbf{w}_k . The error covariance for the predicted state, $\mathbf{P}_{k+1,k}$ is defined as

$$\begin{aligned} \mathbf{P}_{k+1,k} &= E((\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})(\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})^T) \\ &= \mathbf{A}_k E((\mathbf{X}_k - \mathbf{x}_{k,k})(\mathbf{X}_k - \mathbf{x}_{k,k})^T) \mathbf{A}_k^T + E(\mathbf{w}_k \mathbf{w}_k^T) \\ &= \mathbf{A}_k \mathbf{P}_{k,k} \mathbf{A}_k^T + \mathbf{Q}_k, \end{aligned} \quad (7)$$

where $E(\mathbf{w}_k \mathbf{w}_k^T) = \mathbf{Q}_k$ which is called the system noise covariance.

The estimate is a weighted linear combination of the predicted state vector and the observation vector, which is given by

$$\mathbf{x}_{k+1,k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{x}_{k+1,k} + \mathbf{K}_{k+1}\mathbf{z}_{k+1}, \quad (8)$$

where \mathbf{K}_{k+1} is a weight and called Kalman gain. Next, the error covariance for estimated state, $\mathbf{P}_{k+1,k+1}$ is given by

$$\begin{aligned} \mathbf{P}_{k+1,k+1} &= E((\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k+1})(\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k+1})^T) \\ &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})E((\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})(\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})^T) \\ &\quad (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})^T + \mathbf{K}_{k+1}E(\mathbf{v}_{k+1}\mathbf{v}_{k+1}^T)\mathbf{K}_{k+1}^T \\ &\quad + 2(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})E((\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})\mathbf{v}_{k+1}^T)\mathbf{K}_{k+1}^T \\ &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1,k}(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})^T + \mathbf{K}_{k+1}\mathbf{R}_{k+1}\mathbf{K}_{k+1}^T, \end{aligned} \quad (9)$$

where $E(\mathbf{v}_{k+1}\mathbf{v}_{k+1}^T) = \mathbf{R}_{k+1}$ which is called the measurement noise covariance and $E((\mathbf{X}_{k+1} - \mathbf{x}_{k+1,k})\mathbf{v}_{k+1}^T) = 0$ because \mathbf{v}_{k+1} is assumed to be uncorrelated noise.

Now, the Kalman gain needs to be determined by minimizing variation of error or trace of estimation error covariance. Minimization of variation of error or trace of estimation error covariance gives us

$$\begin{aligned} \mathbf{K}_{k+1} &= \min_{\mathbf{K}_{k+1} \in \mathbb{R}^{n_k \times m}} E(\|\mathbf{x}_{k+1,k+1} - \mathbf{X}_{k+1}\|^2 | \mathbf{z}_1, \dots, \mathbf{z}_{k+1}) \\ &= \min_{\mathbf{K}_{k+1} \in \mathbb{R}^{n_k \times m}} \text{trace}(\mathbf{P}_{k+1,k+1}). \end{aligned} \quad (10)$$

The value of \mathbf{K}_{k+1} for which the derivative of $\text{trace}(\mathbf{P}_{k+1,k+1})$ with respect to \mathbf{K}_{k+1} is equal to zero is found from

$$\frac{\partial \text{trace}(\mathbf{P}_{k+1,k+1})}{\partial \mathbf{K}_{k+1}} = -2(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1,k}\mathbf{H}_{k+1}^T + 2\mathbf{K}_{k+1}\mathbf{R}_{k+1} = \mathbf{0}. \quad (11)$$

Then, the Kalman gain, \mathbf{K}_{k+1} is given by

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1,k}\mathbf{H}_{k+1}^T(\mathbf{H}_{k+1}\mathbf{P}_{k+1,k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}. \quad (12)$$

The Kalman filter algorithm is summarized as follows:

$$\begin{aligned} \mathbf{x}_{k+1,k} &= \mathbf{A}_k\mathbf{x}_{k,k} + \mathbf{F}_k \\ \mathbf{P}_{k+1,k} &= \mathbf{A}_k\mathbf{P}_{k,k}\mathbf{A}_k^T + \mathbf{Q}_k \\ \mathbf{K}_{k+1} &= \mathbf{P}_{k+1,k}\mathbf{H}_{k+1}^T(\mathbf{H}_{k+1}\mathbf{P}_{k+1,k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \\ \mathbf{x}_{k+1,k+1} &= \mathbf{x}_{k+1,k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\mathbf{x}_{k+1,k}) \\ \mathbf{P}_{k+1,k+1} &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1,k}. \end{aligned} \quad (13)$$

The detailed discussion and derivation of the Kalman filter can be found in Reference 12.

The last task is to determine \mathbf{Q} and \mathbf{R} , which are system noise covariance and measurement noise covariance, respectively. In general, they are unknown, so they are used as tuning parameters which are determined by trial and error.¹³ Theoretically estimating the system noise covariance \mathbf{Q} and the measurement noise covariance \mathbf{R} is challenging, which is still an unsolved research topic. However, there are several methods for estimating \mathbf{Q} and \mathbf{R} . For example, there are the

maximum likelihood method, the autocovariance least-squares method, and the adaptive sequential estimation method. The main idea of all such methods is that measured data contains sufficient information about unknown noise statistics.

The maximum likelihood method uses probability of measurement \mathbf{z}_k for given \mathbf{Q} and \mathbf{R} because \mathbf{z}_k is the only measurable value. \mathbf{Q} and \mathbf{R} are chosen such that they maximize the cumulative probability of measurements. Since the cost of computing cumulative probability for each pair of \mathbf{Q} and \mathbf{R} is highly expensive, this method is limited for a scalar case.¹⁴

Solving the autocovariance least-squares problem can give us an estimate of \mathbf{Q} and \mathbf{R} because the autocovariance matrix can be written in terms of \mathbf{Q} and \mathbf{R} and also estimated using measurement sequences. Then, the solution of the autocovariance least-squares problem is the estimate of \mathbf{Q} and \mathbf{R} .¹⁵⁻¹⁷

For the adaptive sequential estimation method, a second moment of the approximate system noise and measurement noise over a moving window are computed with an initial guess of \mathbf{Q} and \mathbf{R} and assumption of constant \mathbf{Q} and \mathbf{R} over the moving window. This second moment can also be written in terms of system model parameters and by equating the second moments, \mathbf{Q} and \mathbf{R} are adaptively estimated.¹⁸⁻²²

The maximum likelihood method and the autocovariance least-squares method are not cheap to employ in real-time application. The adaptive sequential estimation method described in Reference 22 takes advantage of a recursive format, making the computation to update noise covariances cheap. However, it still cannot be run in real-time in our numerical example presented in Section 5. Therefore, the noise covariances need to be determined by ad hoc methods or heuristically. Currently, we are developing a reduced data assimilation algorithm such that the adaptive estimation method can be used. The layer-by-layer process results in steady-state under a few layers and the steady-state is reached fast.²³ With this fact, we can divide entire layers into active layers where temperature changes significantly and frozen layers where temperature reaches the steady-state.

In practice, \mathbf{R} is usually determined from the sensor vendor specification or \mathbf{R} can be estimated by measurement reproducibility tests. However, \mathbf{Q} is considered as a tuning parameter in Kalman filter and there are industrial recipes for \mathbf{Q} based on trial and error. In this case, the quality of Kalman filter performance is mainly dependent on engineers' experiences. To exclude such a limitation, one of the optimality tests presented in Reference 24 is used. The optimality is assessed by the whiteness of the given sequence of innovation which is defined as

$$\mathbf{y}_k := \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k,k-1}. \quad (14)$$

Next, we use the property of the autocorrelation function which is defined as

$$\mathbf{L}_i := E(\mathbf{y}_k \mathbf{y}_{k+i}^T). \quad (15)$$

It is known that the autocorrelation function of white noises is zero for nonzero i .²⁵ In the optimality test, let us consider the sequence of N innovation vectors $\mathbf{y}_k, \mathbf{y}_{k+1}, \dots, \mathbf{y}_{k+N-1}$, then the autocorrelation function is estimated by

$$\mathbf{L}_i^* = \frac{1}{\sqrt{N(N-i)}} \sum_{j=0}^{N-i-1} \mathbf{y}_{j+k} \mathbf{y}_{j+k+i}^T, \quad (16)$$

where $1 \leq i \leq N-1$. The optimality metric is computed by summing squares of each diagonal elements of the normalized autocorrelation function which is given by

$$\mathbf{\Gamma}_i^* = \mathbf{L}_i^* / \sqrt{\text{diag}(\mathbf{L}_0) * \text{diag}(\mathbf{L}_i^*)}, \quad (17)$$

using MATLAB notation. This optimality metric gets smaller as the innovations show more whiteness. Thus, we pick \mathbf{Q} such that its optimality is the smallest among tests. For more details, please refer to Reference 24.

3 | MODELING

In this section, a thermal model of the FDM process is discussed. The FDM process modeling is divided into a deposition process and cooling process.

3.1 | Deposition process

We develop a model to emulate deposition of filament during material extrusion process. The deposition process is driven by a G-code which is a series of machine instructions for toolpaths. In a virtual space, the deposition process is modeled as successive addition of voxels as shown in Figure 2. First, a given geometry is discretized into voxels which are rectangular cuboids. Then, the voxels are ordered along the printing path specified by the G-code. This is done by using an in-house Matlab script. Given a list of voxels and their centers, we search along the toolpath over time to determine what voxels are added at any given time step. To facilitate the search, the voxel centers are put into a K-D tree ($k = 3$). With a current print head position as dictated by the G-code, we use a nearest neighbor search to find the nearest neighboring voxel center and add it to the physical domain if the path lies inside the voxel's space.

To ensure that all voxels are added, two extensions to the above K-D tree search are done: (1) intermediate path position searches and (2) greater range searches. In (1) we simply search more positions along the toolpath between each time step as a fast-moving print head may “skip” over voxels with a larger time step size. For (2) if there is a toolpath that travels irregularly over cuboid voxels, for example, traveling in an ellipse, we may skip over voxels with centers slightly outside the path. Instead of only searching for one nearest neighbor to the toolpath, we can find more than one nearest neighbor and add them to the domain if they are within a given distance of the toolpath position. For the geometry described later in Section 5, only (1) was employed as the regular toolpath geometry did not require (2). Note that currently, the model assumes we print with 100% infill.

3.2 | Cooling process

In this section, a cooling process is modeled to compute the thermal field of the part being printed. As the melted filament comes out of the nozzle, it starts to cool down and the previous layer gets hotter due to heat transfer. Assuming no heat generation from mechanical deformation, the cooling process is described by the heat conduction equation which is given by Equation (18)

$$\rho c \dot{T} = \nabla \cdot (k \nabla T), \quad (18)$$

where ρ is the density, c is the specific heat capacity, and T is the temperature. The temperature of the new material that is leaving the nozzle initially is equal to the nozzle temperature. The heat source term is removed from Equation (18) because there is no heat generated. Constant material properties (ρ , c , and k) are assumed. Note that this assumption allows the thermal model to be linear which is an important assumption for the linear Kalman filter application. In addition to that, transversely isotropic material properties are assumed. The plane of isotropy is the plane of the bed plate (xy -plane). Anisotropy originates from weak adhesion between layers along the layer stacking direction (z direction). Thermal conductivity along the z direction is less than along x and y directions. In Section 5, experimental values for acrylonitrile butadiene styrene (ABS) are given.

Equation (18) is solved using a finite difference method. Let us denote Δx , Δy , Δz , and Δt is the grid size in x direction, the grid size in y direction, the grid size in z direction, and the time step size, respectively. For the Laplacian operator, the central difference scheme which is given by Equation (19) is used and for the time integrator, the forward Euler method which is given by Equation (20) is used.

$$\begin{aligned} \nabla^2 T(x, y, z, t) \approx & \frac{T(x + \Delta x, y, z, t) - 2T(x, y, z, t) + T(x - \Delta x, y, z, t)}{\Delta x^2} \\ & + \frac{T(x, y + \Delta y, z, t) - 2T(x, y, z, t) + T(x, y - \Delta y, z, t)}{\Delta y^2} \end{aligned}$$



FIGURE 2 Deposition process modeling

$$+ \frac{T(x, y, z + \Delta z, t) - 2T(x, y, z, t) + T(x, y, z - \Delta z, t)}{\Delta z^2}. \quad (19)$$

$$\dot{T}(x, y, z, t) \approx \frac{T(x, y, z, t + \Delta t) - T(x, y, z, t)}{\Delta t}. \quad (20)$$

The forward Euler method requires a stability condition on time step size. Based on Fourier analysis, the stability condition is given by Equation (21)

$$\alpha \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) \leq 1/2, \quad (21)$$

where α is given by $\frac{k}{\rho c}$.²⁶ This condition is readily satisfied because the extrusion process occurs quickly, and this requires a smaller time step size than the stability condition. Numerical values for our model are shown in Section 5.

For boundary conditions, convection and radiation are applied to the surfaces. To make the model linear, radiation is treated as convection using the following equation

$$\epsilon \sigma (T^4 - T_\infty^4) \approx 4\epsilon \sigma T_{\text{mean}}^3 (T - T_\infty) = h_r (T - T_\infty), \quad (22)$$

where ϵ is emissivity, $T_{\text{mean}} = \frac{T + T_\infty}{2}$, and $h_r = 4\epsilon \sigma T_{\text{mean}}^3$.

Writing the discretized equation into matrix form yields

$$\mathbf{T}^{n+1} = \mathbf{A}^n \mathbf{T}^n + \mathbf{F}^n, \quad (23)$$

where n denotes time step. In Equation (23), \mathbf{A} and \mathbf{F} are the system models that are going to be used in the linear Kalman filter algorithm. Both the values and the sizes of the system models, \mathbf{A} and \mathbf{F} , vary because the domain and the boundary surfaces evolve as the material is deposited.

4 | DATA ASSIMILATION

For data assimilation, the linear Kalman filter is used. To run the Kalman filter algorithm, the system models, observational models, system noise covariance, and measurement noise covariance, which are represented by \mathbf{A} , \mathbf{F} , \mathbf{H} , \mathbf{Q} , and \mathbf{R} , respectively are needed.

By discretizing the governing equation and writing it in matrix form, the system models which are denoted as \mathbf{A} and \mathbf{F} are included in the Kalman filter. IR camera measurements determine the observation model which is represented by \mathbf{H} . The system noise covariance, \mathbf{Q} and the measurement noise covariance, \mathbf{R} are determined by the optimality test.²⁴

The linear Kalman filter algorithm is revisited and given by:

$$\begin{aligned} \mathbf{x}_{k+1,k} &= \mathbf{A}_k \mathbf{x}_{k,k} + \mathbf{F}_k \\ \mathbf{P}_{k+1,k} &= \mathbf{A}_k \mathbf{P}_{k,k} \mathbf{A}_k^T + \mathbf{Q}_k \\ \text{if } &\text{measurement occurs} \\ \mathbf{K}_{k+1} &= \mathbf{P}_{k+1,k} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1,k} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \\ \text{if } &\text{no measurement occurs} \\ \mathbf{K}_{k+1} &= \mathbf{0} \\ \mathbf{x}_{k+1,k+1} &= \mathbf{x}_{k+1,k} + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \mathbf{x}_{k+1,k}) \\ \mathbf{P}_{k+1,k+1} &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1,k}. \end{aligned} \quad (24)$$

IR camera

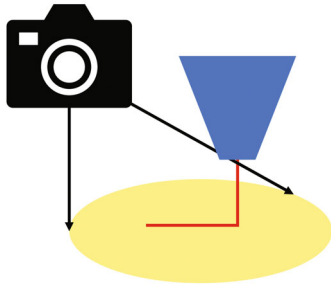


FIGURE 3 Virtual experimental setup

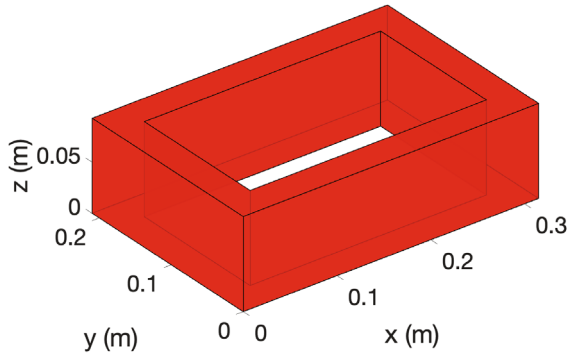


FIGURE 4 Geometry for numerical test

The computational complexity of this algorithm is $O(n^3)$, where n is the size of state vector, \mathbf{x} due to matrix multiplication in $\mathbf{A}\mathbf{P}\mathbf{A}^T$. Since n is increasing as a new material is deposited, it is expensive to use the Kalman filter algorithm expressed in Equation (24). Fortunately, the Kalman gain \mathbf{K} can be precomputed offline because only the calculation steps for error covariance matrix \mathbf{P} are required, which can run without observation data \mathbf{z} . We also know in advance the system model \mathbf{A} , the initial guess of error covariance $\mathbf{P}_{0,0}$, and the noise covariances. The observation matrix \mathbf{H} is determined from the sensor configuration which can be given by a preset path or updated adaptively based on the error covariance matrix \mathbf{P} in our work. With the precomputed Kalman gain, the data assimilation process is completed in two steps as follows:

$$\mathbf{x}_{k+1,k} = \mathbf{A}_k \mathbf{x}_{k,k} + \mathbf{F}_k$$

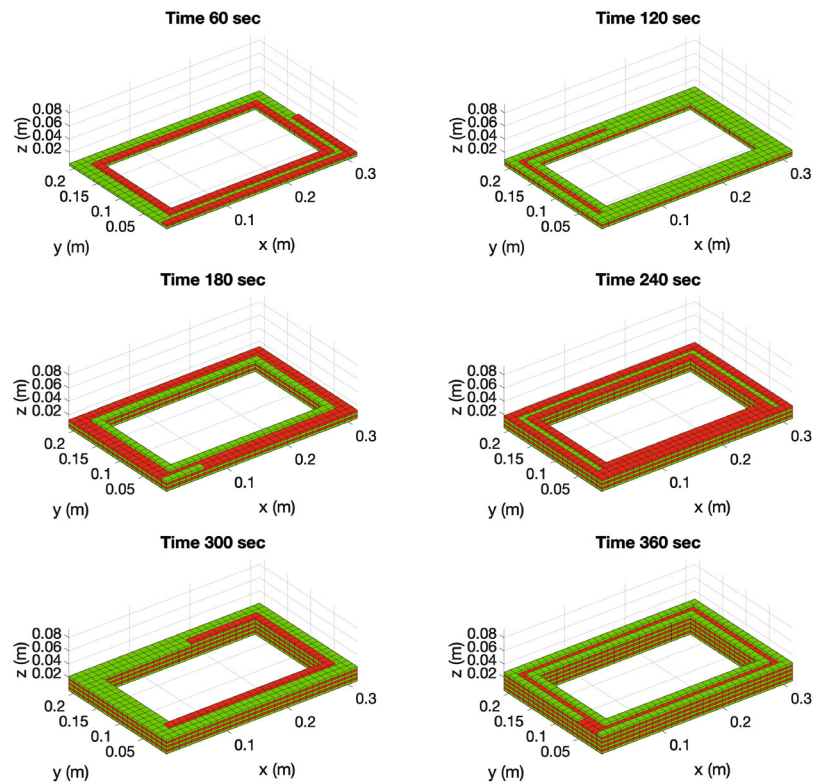
$$\mathbf{x}_{k+1,k+1} = \mathbf{x}_{k+1,k} + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \mathbf{x}_{k+1,k}). \quad (25)$$

The computational complexity of this step is $O(n^2)$ when \mathbf{A} is dense and $O(n)$ when \mathbf{A} is sparse. In our work, the system model \mathbf{A} is sparse because the Laplacian term in the heat equation is approximated using only adjacent grid points and explicit time schemes such as the forward Euler method enable us to use the Laplacian operator as the system model \mathbf{A} for time marching. In the case of implicit schemes like the backward Euler method, we need to solve a system of equations at each time step, which make the system model \mathbf{A} nonsparse because of the inverse operation involving the Laplacian operator. With the sparsity of matrix \mathbf{A} , the complexity reduces to $O(n)$, which is much less expensive than $O(n^3)$.

5 | RESULTS AND DISCUSSION

A numerical experiment has been conducted to verify the performance of the proposed data assimilation algorithm. The virtual experimental setup is shown in Figure 3.

We assume that an IR camera observes the top surface temperature at every time step. The geometry used for the test is shown in Figure 4. The STL file of this geometry is sliced and then the G-code is generated using the open source slicer, Slic3r. After voxelizing the given geometry, the in-house Matlab script is used to put each voxel in printing order as shown in Figure 5. Then, discretizing the heat equation with given voxels yields the system models for the linear Kalman filter.

FIGURE 5 Voxel ordering**TABLE 1** Printing parameters

Nozzel Temperature, T_n	473.15K
Bed Temperature, T_b	323.15K
Ambient Temperature, T_a	298.15K
Heat Transfer Coefficient, h	8.5W/m ²
Thermal Conductance, h_c	1000W/m ² ·K
Nozzle Speed, V_n	70mm/s

TABLE 2 Material parameters

Density, ρ	1100kg/m ³
Heat capacity, c	1740J/kg·K
Thermal conductivity in xy direction, k_{xy}	1.2W/m·K
Thermal conductivity in z direction, k_z	0.3W/m·K
Emissivity, e	0.9

The virtual ground truth data and measured data are generated by adding random noise to numerical simulation results. The thermal field computed with data assimilation using Kalman filter and without the data assimilation algorithm are compared with the virtual ground truth data.

Time step size, grid size in x direction, grid size in y direction, and grid size in z direction are $\Delta t = 0.15$ seconds, $\Delta x = 10.5$ mm, $\Delta y = 10.5$ mm, and $\Delta z = 3.1$ mm, respectively. The system noise covariance is set to 10 for nodal positions on the top surface and set to 1 for the others because more uncertainty exists on the top surface. The measurement noise covariance is set to 10^{-6} .

In this example, large scale printing is assumed, and the numerical test parameters are presented in Tables 1 and 2.

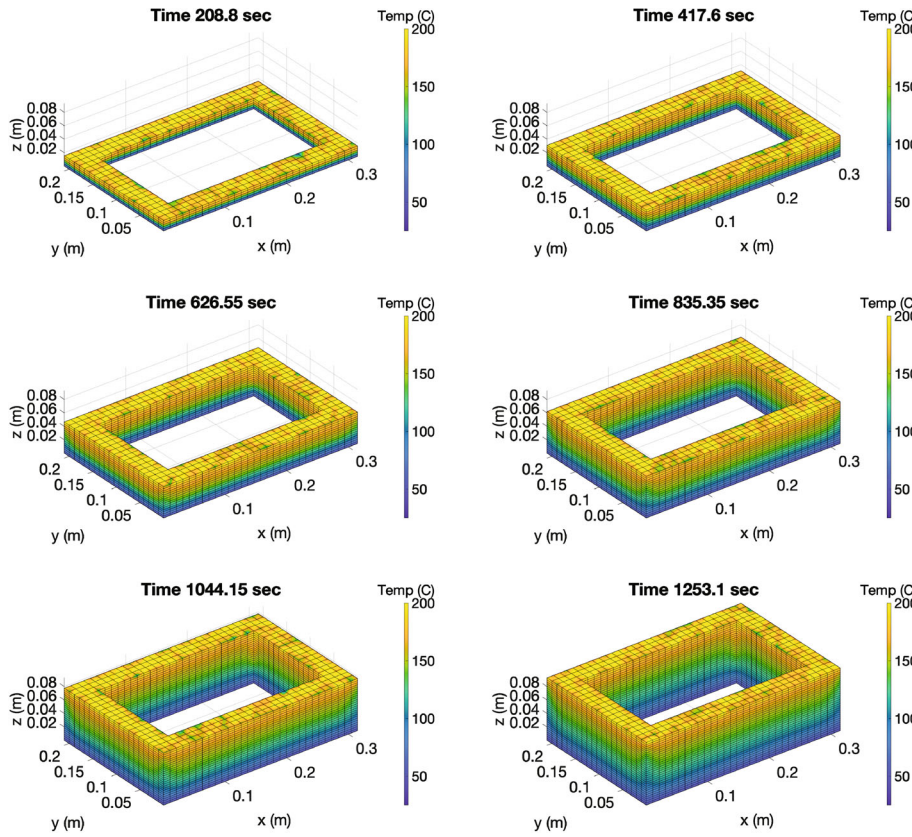


FIGURE 6 Snapshots of estimated thermal field

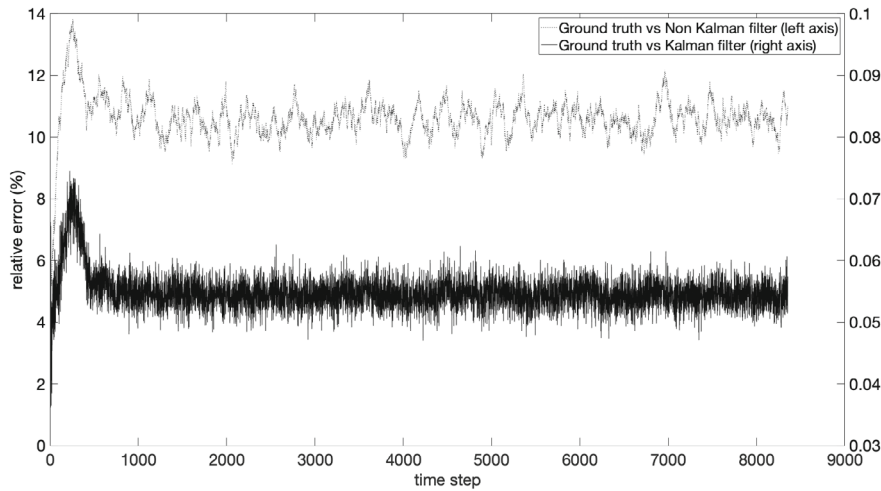


FIGURE 7 Relative errors at time step k of tests with and without using the Kalman filter

With the given parameters, the stability condition for forward Euler time stepping scheme is checked in Equation (26).

$$\frac{k_{xy}}{\rho c} \Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) = 0.0029 \leq 0.5. \quad (26)$$

Since the stability condition is satisfied, the chosen time step can be used.

Figure 6 shows snapshots of the estimated thermal field using the linear Kalman filter data assimilation algorithm.

Figure 7 compares the relative errors at time step k of the top surface temperature field computed by only numerical simulation vs using the data assimilation algorithm.

FIGURE 8 Relative errors at time step k of tests with and without using the Kalman filter with larger R and sparser measurements

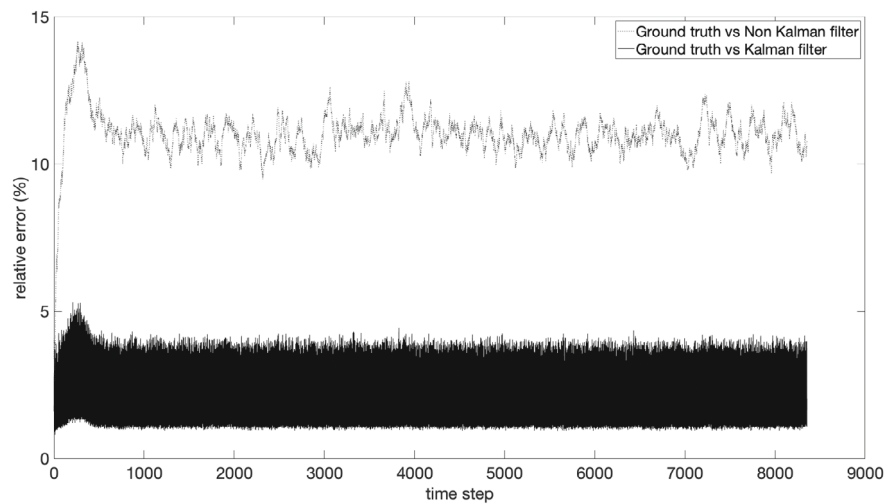


TABLE 3 Optimality test

Q	Optimality metric ($\times 1e^{-5}$)	E
$Q = Q_{\text{true}}$	4.8026	2.8767%
$Q = 10Q_{\text{true}}$	4.8089	2.8864%
$Q = 0.1Q_{\text{true}}$	4.9428	3.1231%

From this result, it is shown that fusing numerical simulation data and measured data significantly improves the accuracy of monitoring the thermal field during AM processes. However, it is not surprising to have such huge improvement in the error because very accurate measurement data is provided. Thus, we also present a more realistic numerical result tested with larger measurement noise covariance (amplitude of measurement noise covariance is 4) and sparser observations (frequency of measurement is 1.1 Hz). Figure 8 compares the relative errors at time step k of the top surface temperature field computed by only numerical simulation versus using the data assimilation algorithm.

From this result, it is shown that fusing numerical simulation data and measured data improves accuracy of monitoring of thermal field during AM processes. Adhesion between layers is highly sensitive to interfacial temperature²⁷⁻²⁹ so, estimating the interfacial temperature accurately using the Kalman filter will help us to qualify the adhesion in printed parts.

As we discussed earlier, the system noise covariance is a tuning parameter in the Kalman filter algorithm. To find the optimal system noise covariance, the optimality metrics presented in²⁴ are computed for Q_{true} , $0.1Q_{\text{true}}$, and, $10Q_{\text{true}}$, where $N = 1364$ innovations are used. In Table 3, the computed values are presented, and we can confirm that when Q is equal to Q_{true} , the optimality metric is the smallest. Moreover, to access the performance of the Kalman filter for each case of Q , we compute the l_2 norm of the top surface state error which is given by

$$E = \frac{\sqrt{\sum_k \|\theta_k - \theta_k^*\|_2^2}}{\sqrt{\sum_k \|\theta_k^*\|_2^2}} \times 100, \quad (27)$$

where $\theta_k \in \mathbb{R}^{n_k^{\text{top}}}$ is the a posteriori temperature and $\theta_k^* \in \mathbb{R}^{n_k^{\text{top}}}$ is the ground truth temperature at time step k . Here, n_k^{top} is the number of degrees of freedom that can be viewed from the top at time step k . We can also see that the E is the smallest when Q is equal to Q_{true} .

In practice, Q_{true} is not known, so we may pick Q such that its optimality is the smallest among several candidates for Q .

Since the Kalman filter outputs estimation error covariance, or uncertainty of estimation, this value can be used to guide the positions of IR cameras. In the case of large-scale 3D printing, multiple IR cameras may be required to monitor the process. These IR cameras can then be guided to observe the location where uncertainty is high, resulting in a closed-loop monitoring system.

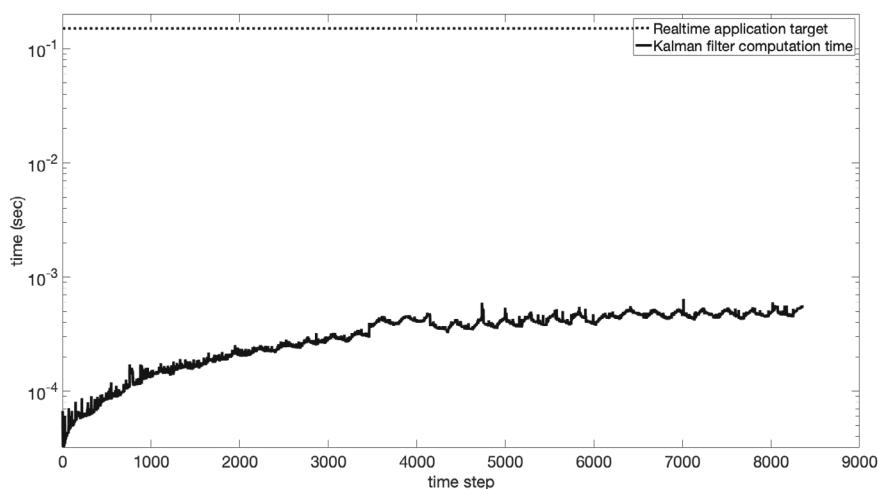


FIGURE 9 Computation time for each time step

Furthermore, to be included in the real-time feedback control system, computation time for each voxel should be faster than physical extrusion time for the same amount of material. To verify this, the numerical test where data assimilation is done every time step was conducted on our local machine equipped with 3 GHz CPU and 64 GB RAM. Note that the implementation was not parallelized, so it ran on a single core. Figure 9 shows the computation time for every time step. The computation time was well below the real-time application target. The real-time application target was set to 0.15 seconds because it takes 0.15 seconds to extrude the same amount of material as one voxel. Therefore, since we are able to compute the full thermal field within this limit, use of a real-time feedback control loop for this process is feasible. At the final step, the number of voxels was 7920 and the computation time was $5.15 \cdot 10^{-4}$ seconds.

As we discussed in Section 4, computational complexity is $O(n)$. Therefore, the data assimilation algorithm can be used for real-time application up to $2.31 \cdot 10^6$ voxels. The total volume of extruded material in that case would be 0.79 m^3 and the corresponding mass, in the case of ABS as tested in this paper, would be 867.67 kg. For example, Oak Ridge National Laboratory printed a full-size car frame using their big area additive manufacturing system which had a volume of approximately 0.1 m^3 .^{30,31} This is within the real-time limits, so the data assimilation algorithm could even be used for large components similar to the car frame.

6 | CONCLUSION

In this paper, a data assimilation technique using the linear Kalman filter is developed for monitoring extrusion-based AM processes. It is applicable to any type of AM processes, provided we can observe the thermal field of parts as they are being processed. The developed method takes into account both numerical simulation data from the physical model and measured data from sensors. A virtual experiment was performed and showed the accuracy of the estimation of the thermal field improved using the presented method. Furthermore, uncertainty values output by the Kalman filter can be employed to guide sensors, resulting in a closed-loop monitoring process. This method shows the potential to be utilized in real-time feedback control systems for AM processes to improve the print quality.

ACKNOWLEDGEMENTS

This work was supported by the Advanced Robotics for Manufacturing (ARM 17-01-F-C07). We would like to acknowledge Professor Zohdi for his guidance. We also would like to express special thanks to our colleague, Zachary Yun, for his helpful discussion.

ORCID

Youngkyu Kim  <https://orcid.org/0000-0002-4825-4072>

REFERENCES

1. Turner Brian N, Robert S, Gold Scott A. A review of melt extrusion additive manufacturing processes: I. process design and modeling. *Rapid Prototyp J.* 2014;20(3):192-204. <http://dx.doi.org/10.1108/rpj-01-2013-0012>.

2. Ahn S-H, Montero M, Odell D, Roundy S, Wright PK. Anisotropic material properties of fused deposition modeling ABS. *Rapid Prototyp J*. 2002;8(4):248-257.
3. Seppala JE, Migler KD. Infrared thermography of welding zones produced by polymer extrusion additive manufacturing. *Addit Manuf*. 2016;12:71-76.
4. Kishore V, Ajinjeru C, Nycz A, et al. Infrared preheating to improve interlayer strength of big area additive manufacturing (BAAM) components. *Addit Manuf*. 2017;14:7-12.
5. Kalman RE. A new approach to linear filtering and prediction problems. *J Basic Eng*. 1960;82(1):35-45.
6. Kalman RE, Bucy RS. New results in linear filtering and prediction theory. *J Basic Eng*. 1961;83(1):95-108.
7. Eleffendi MA, Johnson CM. Application of Kalman filter to estimate junction temperature in IGBT power modules. *IEEE Trans Power Electron*. 2015;31(2):1576-1587.
8. Bertino L, Evensen G, Wackernagel H. Sequential data assimilation techniques in oceanography. *Int Stat Rev*. 2003;71(2):223-241.
9. Cha Y-J, Chen JG, Büyüköztürk O. Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters. *Eng Struct*. 2017;132:300-313.
10. Julier SJ, Uhlmann JK, Durrant-Whyte HF. A new approach for filtering nonlinear systems. Paper presented at: Proceedings of 1995 American Control Conference-ACC'95; vol. 3, 1995:1628-1632; Seattle, WA, USA, USA: IEEE.
11. Rao CR. Information and the accuracy attainable in the estimation of statistical parameters. *Breakthroughs in Statistics*. New York, NY: Springer; 1992:235-247.
12. Reid I, Term H. *Estimation Lecture Notes*. Vol ii. Oxford, UK: University of Oxford; 2001.
13. Kim P. *Kalman Filter for Beginners: With MATLAB Examples*. Scotts Valley, CA: CreateSpace; 2011.
14. Matisko P, Havlena V. Noise covariances estimation for Kalman filter tuning. *IFAC Proc Vol*. 2010;43(10):31-36.
15. Odelson BJ, Rajamani MR, Rawlings JB. A new autocovariance least-squares method for estimating noise covariances. *Automatica*. 2006;42(2):303-308.
16. Rajamani MR, Rawlings JB, Soderstrom TA. Application of a new data-based covariance estimation technique to a nonlinear industrial blending drum. *Texas-Winsconsin Modeling and Control Consortium, Tech. Report*. 2007; 3.
17. Lima FV, Rawlings JB. Nonlinear stochastic modeling to improve state estimation in process monitoring and control. *AIChE J*. 2011;57(4):996-1007.
18. Myers K, Tapley B. Adaptive sequential estimation with unknown noise statistics. *IEEE Trans Autom Control*. 1976;21(4):520-523.
19. Sage AP, Husa GW. Algorithms for sequential adaptive estimation of prior statistics. Paper presented at: Proceedings of the 1969 IEEE Symposium on Adaptive Processes (8th) Decision and Control; 1969:61; IEEE; University Park, PA.
20. Wang X, Zhao X, Yang X. Research of motion resolving and filtering algorithm of a ship's three-freedom motion simulation platform based on LabVIEW. *J Phys Conf Ser*. 2006;48:149.
21. Narasimhappa M, Rangababu P, Sabat SL, Nayak J. A modified sage-husa adaptive kalman filter for denoising fiber optic gyroscope signal. Paper presented at: Proceedings of the 2012 Annual IEEE India Conference (INDICON); 2012:1266-1271; Kochi, **India**, IEEE.
22. Akhlaghi S, Zhou N, Huang Z. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. Paper presented at: Proceedings of the 2017 IEEE Power & Energy Society General Meeting; 2017:1-5; Chicago, IL, USA: IEEE.
23. Compton BG, Post BK, Duty CE, Love L, Kunc V. Thermal analysis of additive manufacturing of large-scale thermoplastic polymer composites. *Addit Manuf*. 2017;17:77-86.
24. Matisko P, Havlena V. Optimality tests and adaptive Kalman filter. *IFAC Proc Vol*. 2012;45(16):1523-1528.
25. A. Papoulis, Probability, Random Variables, and Stochastic Processes. McGraw-Hill New York, NY 19842 (1991) 345-348.
26. LeVeque RJ. Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. *Society for Industrial and Applied Mathematics*. Vol 98. Philadelphia, PA: Siam; 2007.
27. Coogan TJ, Kazmer DO. Bond and part strength in fused deposition modeling. *Rapid Prototyp J*. 2017;23(2):414-422.
28. Sun Q, Rizvi G, Bellehumeur C, Gu P. Effect of processing conditions on the bonding quality of FDM polymer filaments. *Rapid Prototyp J*. 2008;14(2):72-80.
29. Bellehumeur C, Li L, Sun Q, Gu P. Modeling of bond formation between polymer filaments in the fused deposition modeling process. *J Manuf Process*. 2004;6(2):170-178.
30. Holshouser C, Newell C, Palas S, et al. Out of bounds additive manufacturing. *Adv Mater Process*. 2013;171(3):15-17.
31. Curran S, Chambon P, Lind R, et al. Big area additive manufacturing and hardware-in-the-loop for rapid vehicle powertrain prototyping: A case study on the development of a 3-D-printed Shelby Cobra. *SAE Technical Paper Series*, 2016; <https://doi.org/10.4271/2016-01-0328>.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Kim Y, Alcantara D, Zohdi TI. Thermal state estimation of fused deposition modeling in additive manufacturing processes using Kalman filters. *Int J Numer Methods Eng*. 2022;123:3021-3033. <https://doi.org/10.1002/nme.6490>