# Enabling In-Time Prognostics with Surrogate Modeling through Physics-Enhanced Dynamic Mode Decomposition Method

Katelyn Jarvis[1], Matteo Corbetta[2], Christopher Teubert[3], and Stefan Schuet[4]

[1,3,4] *NASA Ames Research Center, Moffett Field, CA 94035*
*katelyn.j.jarvis@nasa.gov*
*christopher.a.teubert@nasa.gov*
*stefan.r.schuet@nasa.gov*

[2] *KBR Inc., NASA Ames Research Center, Moffett Field, CA 94035*
*matteo.corbetta@nasa.gov*

## Abstract

Computational models provide essential quantitative tools for assessing and predicting the health and performance of physical systems. However, high-fidelity models are rarely used in real-time operations or large optimization loops, due to their time-intensive nature. A common approach to improving computational efficiency of prognosis is to employ surrogate models. Such models can significantly decrease computation time for some accuracy loss. In this context, use of Dynamic Mode Decomposition (DMD) is proposed to generate surrogate models for lithium-ion (Li-ion) battery discharge. DMD has been suggested and used successfully in the area of fluid dynamics for over a decade, but it has not been applied to the Prognostics and Health Management domain, where far-ahead prediction of nonlinear behavior is crucial to propagate faults or predict Remaining Useful Life (RUL). For Li-ion battery health management, the standard application of DMD using only the observable quantities of interest was unable to capture the nonlinear discharge of batteries exhibited in lab testing. A potential solution was found by implementing Koopman theory, which considers the dynamics of nonlinear systems. Koopman theory provides a mechanism to trade-off low dimensional nonlinear models with high-dimensional linear ones in a DMD framework, by augmenting nonlinear state variables into the system representation. For battery health management, we augmented the observable variables with the hidden states of a higher-fidelity physics model to build the DMD surrogate. In comparison to a high-fidelity model, the surrogate improved computational efficiency with only a minimal loss of accuracy, and enabled long-term prognostics horizons. A generalized method for this was implemented in the `ProgPy` python packages.

## 1. Motivation

The field of prognostics and health management (PHM) involves developing quantitative tools to estimate the health of physical systems and predict how system degradation will evolve with use. To aid in risk monitoring and mitigation, prognostics algorithms must be both accurate and timely. Often, high accuracy is attained by representing physical systems with high-fidelity models, which can be computationally expensive. For example, in some resource-constrained scenarios, such as small Uncrewed Aerial Systems (sUAS), small satellites, or spacecraft, the computational cost of high-fidelity models is prohibitive to achieve real-time or quasi real-time prognostics. This is especially true in cases where prognostics is performed inside an optimization loop, such as an application where system loading is optimized to maximize useful life.

Surrogate, or reduced-order, modeling is a computationally efficient alternative to more detailed physical models. In surrogate modeling, a lower-fidelity, faster version of the model (i.e., the surrogate) is generated and used in place of the original high-fidelity model. This approach can be used to speed up complex simulations when the need for faster prediction outweighs the need for high accuracy. For example, in simulations of advanced air mobility vehicles, surrogate models of the battery and powertrain can provide an alternative to physical models that describe the full electrochemistry of the battery and the high-frequency electronic speed controller, leading to faster simulations when highly detailed modeling is not necessary to assess the system performance.

One such surrogate modeling technique is Dynamic Mode Decomposition (DMD), a powerful data-driven tool used to identify approximate linear dynamics from high-dimensional data (Kutz, Brunton, Brunton, & Proctor, 2016). Developed originally within the computational fluid dynamics community by Schimd (Schmid, 2010), DMD and its variants have been successfully used for more than a decade. The method is remarkably simple and requires minimal assumptions about the underlying physical system. In its standard form, DMD aims at approximating the nonlinear dynamics of a system through a linearized model expressed as

$$X' = AX\,, \tag{1}$$

where $X$ is a "snapshot" matrix, with each column consisting of a vector of observable data from the system at successive points in time (i.e. column one includes all observable data at $t_0$, column two includes data at $t_1$, etc.). Similarly, $X'$ contains the same observable data one step forward in time (i.e. from $t_1$ onward; see Section 2.1 for details). Once constructed, the linear approximation can be solved for by a matrix pseudo-inversion, and produces accurate look-ahead predictions for short time horizons (Kutz et al., 2016). Its simplicity is also supported by the computational cost, limited to the computation time of a single Singular-Value Decomposition (SVD) for the regression stage. However, while proven to be a valuable tool in specific applications for short time scales, of the order of a few time steps $\Delta t$, DMD has yet to be explored extensively in a prognostics setting where the system state dimension is not always large, fast computations can be crucial, and long prediction horizons are necessary. Long prediction horizons extend, for the sake of this work, to hundreds or thousands of time steps $\Delta t$ beyond of the initial condition. They differ from short prediction horizons, for which DMD has proven successful, where only a few time steps ahead are predicted and long-term dynamics can be ignored. In the test case we present in this paper, the prediction horizon is of the order of $10^3$ s, e.g., $\sim$1000 to $\sim$3000 s, with $\Delta t$ of the orders of $\sim\ 10^{-1}$s to $\sim\ 10^{1}$s.

As motivation for using DMD in a prognostics scenario, we performed some exploratory work using DMD to predict the voltage discharge curve of a lithium-ion (Li-ion) battery subject to step-wise randomized current input. The discharge curve was generated using a well-established, nonlinear electrochemistry model (Daigle & Kulkarni, 2013), which assumes the state of the battery health is known and describes the nominal voltage discharge that results from a current imposed upon the battery. The relationship between input current and output voltage is highly nonlinear, and thus provides an ideal opportunity for DMD to approximate this complex system with a linear model.

As in a standard DMD application, we construct a snapshot matrix with only output quantities. However, in the proposed battery case study, there is just one output quantity (i.e. voltage) at each point in time. As in previous work with limited state dimensions (Tirunagari, Kouchaki, Poh, Bober, & Windridge, 2017), we attempted to adjust for this by composing each column of the snapshot matrix with a set of lagged time series of voltage values, similarly to an auto-regressive model. For configurable model order, $j$, $X$ is:

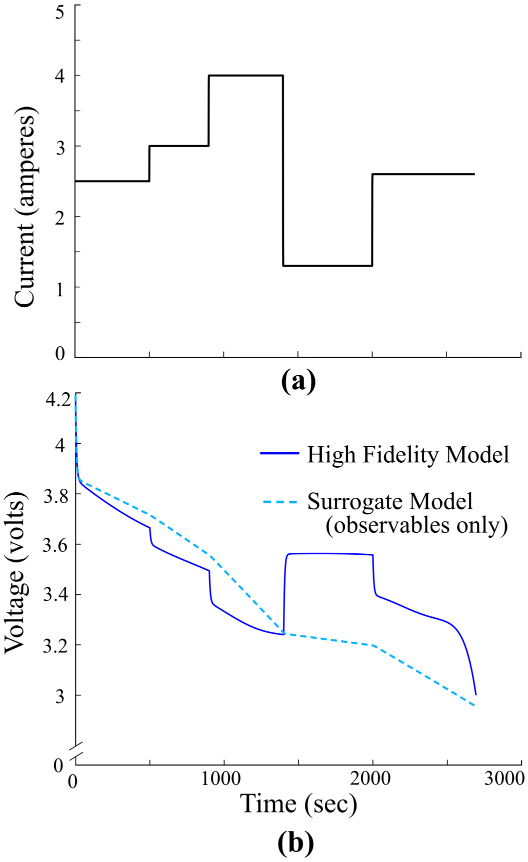

Figure 1. Prediction of a voltage discharge curve (b) given a current loading profile (a) using DMD with one-dimensional, lagged time series of voltage values.

$$X = \begin{bmatrix} V(t_j) & V(t_{j+1}) & \dots & V(t_{j+(m-1)}) \\ \vdots & \vdots & & \vdots \\ V(t_0) & V(t_1) & \dots & V(t_{m-1}) \end{bmatrix} \tag{2}$$

(Also see Eq. (1) in (Tirunagari et al., 2017)).

However, applying DMD in this way, the model is unable to capture the nonlinear dynamics of voltage discharge, thus failing to produce acceptable predictions (Figure 1).

This poor linearization forced us to explore alternatives to the use of observable states only. An extension of DMD was presented in (Williams, Kevrekidis, & Rowley, 2015), where the authors proposed to augment the set of observables with a set of basis functions that would aid the regression. By doing so, the increased number of rows in the snapshot matrices increases the dimensionality of the model and its ability to reproduce the reduced-order dynamics (this approach connects DMD with Koopman theory (Koopman, 1931)). However, the selection of the basis functions is arbitrary, and there is no guarantee that the functions selected for one application will work seamlessly for others. For the same battery case study, we tried using radial basis functions of type "thin plate spline" following the approach in (Korda & Mezić, 2018), where one hundred basis functions with random control points were used to augment the dimension of the observable states. Korda and Mezić successfully found a finite-dimensional approximation of the Koopman operator for the nonlinear systems they studied, and applied it to model a predictive control problem. Their use-case required a limited number of predictions steps, given the high-frequency nature of feedback controls. In our case, however, the success encountered in short-term state forecasts did not translate to accurate predictions of a full voltage discharge curve, regardless of our many attempts to tune the model by changing the type and number of basis functions.

In this work, we explore an alternative extension of DMD that utilizes the internal states and equations of a high-fidelity model of the system. Exploiting such a model (as in the electrochemistry model of the Li-ion battery) provides access to unobservable system state information, and we augment our snapshot matrices with this information. This provides the opportunity to enhance the DMD framework with physics-based knowledge (instead of using basis functions that are not driven by physical processes, as in (Williams et al., 2015) and (Korda & Mezić, 2018) above). We define this DMD extension as "physics-enhanced DMD". We find that such a physics-enhanced DMD allowed us to accurately capture battery voltage discharge predictions over a number of blind tests. Thus, this work highlights the potential for using physics-enhanced DMD for long-term predictions.

The rest of the paper is structured as follows: Section 2 summarizes DMD and the basic equations necessary to implement the algorithm. Section 3 shows the application of the physics-enhanced DMD as a surrogate for a simple toy problem (Section 3.1) and voltage discharge predictions generated through the electrochemistry battery model (Sections 3.2 - 3.5). Section 4 concludes the paper with a critical analysis and suggestions for future work.

## 2. Background

This section summarizes the fundamentals of the DMD algorithm, its connection with the Koopman operator, the SVD-based approach to find matrix $A$ (Eq. (1)) in large scale problems, and the application of DMD with control inputs. Most of the material was retrieved from (Kutz et al., 2016; Williams et al., 2015), and the interested reader is referred to those publications for more details.

We define the discrete dynamical system similarly to (Kutz et al., 2016) as follows:

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{F}(\boldsymbol{x}_k)\,,\\ \boldsymbol{y}_{k+1} &= \boldsymbol{g}(\boldsymbol{x}_{k+1})\,,\end{aligned}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n\times 1}$ is the vector of states representing the dynamical system at time $t_k \in \mathbb{R}^{\geq 0}$, $k \in \mathbb{N}$. Vector $\boldsymbol{y}_k \in \mathbb{R}^{q\times 1}$ represents the measurement vector. Symbols $\boldsymbol{F}(\cdot) : \mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1}$ and $\boldsymbol{g}(\cdot) : \mathbb{R}^{n\times 1} \to \mathbb{R}^{q\times 1}$ represent the state-space functions.

It should be noticed that, in the DMD context, the measurement function $\boldsymbol{g}(\cdot)$ may be composed of equations derived from physical relationships between the hidden states and the "observable" $\boldsymbol{y}$ values, and/or data-driven kernels (such as basis functions) that increase the state vector to a higher dimensionality (Korda & Mezić, 2018), so that DMD can be applied to this higher-dimensional vector $\boldsymbol{y}$ to find a higher-dimensional linearized system (see Section 2.2 for more details).

### 2.1. Basics of DMD: Estimating Matrix $A$

As described in (Kutz et al., 2016), DMD is "an equation-free method capable of providing a decomposition of a complex system into spatiotemporal coherent structures that may be used for short-time future state prediction and control" (chapter 1, page 1). One of the key advantages of DMD is its simplicity. The algorithm starts by using the observable variables of the system to construct snapshot matrices $X$ and $X'$, with $X'$ lagged by one time step, $\Delta t$:

$$X(t) = \begin{bmatrix} x_1(t_0) & \dots & x_1(t_i) & \dots & x_1(t_{m-1}) \\ x_2(t_0) & \dots & x_2(t_i) & \dots & x_2(t_{m-1}) \\ \dots & \dots & \dots & \dots & \dots \\ x_n(t_0) & \dots & x_n(t_i) & \dots & x_n(t_{m-1}) \end{bmatrix}\,,$$

$$X'(t) = \begin{bmatrix} x_1(t_1) & \dots & x_1(t_{i+1}) & \dots & x_1(t_m) \\ x_2(t_1) & \dots & x_2(t_{i+1}) & \dots & x_2(t_m) \\ \dots & \dots & \dots & \dots & \dots \\ x_n(t_1) & \dots & x_n(t_{i+1}) & \dots & x_n(t_m) \end{bmatrix}\,.$$

Thus $X,\, X' \in \mathbb{R}^{n\times m}$. For simplicity of notation, we will describe the dynamical system in its discrete form, omitting the time dependency of the matrices. In the case of unforced systems, DMD aims at finding the constant matrix $A$ to satisfy equality $X' \approx A\,X$, and a best fit can be obtained by:

$$A = X'\,X^{+}\,, \tag{3}$$

where $^{+}$ denotes the inverse (we use the Moore-Penrose pseudo-inverse), and $A \in \mathbb{R}^{n\times n}$. This is equivalent to minimizing the Frobenius norm of the residuals $X' - A\,X$, (Kutz et al., 2016). Once $A$ is computed, the dynamics can then be approximated with discrete first order integration $\boldsymbol{x}_{k+1} \approx A\,\boldsymbol{x}_k$.

### 2.2. Connection with Koopman Operator

DMD attempts to approximate a nonlinear system with a linear one. It can be viewed as a finite-dimensional approximation of the Koopman operator, $\mathcal{K}$ (Koopman, 1931; Kutz et al., 2016), which is an infinite-dimensional, composition operator for $g$ and $\boldsymbol{F}$:

$$\begin{aligned}\mathcal{K}\,g &= g \circ \boldsymbol{F}\,,\\ \mathcal{K}\,g(\boldsymbol{x}_k) &= g(\boldsymbol{F}(\boldsymbol{x}_k)) = g(\boldsymbol{x}_{k+1})\,.\end{aligned}$$

As stated in (Kutz et al., 2016), this is different from linearizing the system dynamics, where it is necessary to define an equilibrium point, $\boldsymbol{x}_0$, around which the linearization is valid. In the next section, we will utilize this principle by introducing hidden states of the physical model as observables to build matrix $A$.

### 2.3. DMD for Large-Scale Problems

DMD was originally developed for use in the fluid dynamics community and is routinely used on large scale problems (Kutz et al., 2016). In these cases, where the state dimension $n$ is on the order of thousands, the inversion of $X$ can be numerically intractable, thus requiring rank reduction to build the matrix $A$, such as singular-value decomposition (SVD) (Stewart, 1993). While the problems presented here do not require this (the electrochemistry model has $n = 8$ states), the SVD-based algorithm is common in solving DMD, and thus we provide a brief summary. The interested reader is referred to (Kutz et al., 2016) for more details.

Matrix $X$ is first split into $U$, $\Sigma$, $V^*$ according to SVD, where $U$ and $V$ are the left and right eigenvectors of $X$, respectively, $\Sigma$ is a square diagonal matrix of eigenvalues, and superscript $^*$ indicates the complex conjugate. Matrix $A$ is calculated using the pseudo-inverse of $X$ through SVD:

$$A = X'\,V\,\Sigma^{-1}\,U^*\,. \tag{4}$$

To reduce dimensionality, matrices $U$, $\Sigma$, $V$ are truncated at the desired reduced rank $r$ such that $\tilde{U} \in \mathbb{C}^{n\times r}$, $\tilde{\Sigma} \in \mathbb{C}^{r\times r}$, $\tilde{V} \in \mathbb{C}^{m\times r}$. The rank-reduced representation of $A$, $\tilde{A}$, can be obtained from Eq. (4) by pre- and post-multiplying $A$ by $U^*$ and $U$, respectively, which leads to:

$$\tilde{A} = \tilde{U}^*\,X'\,\tilde{V}\,\tilde{\Sigma}^{-1}\,,$$

thus obtaining the reduced-rank dynamical system: $\tilde{\boldsymbol{x}}_{k+1} = \tilde{A}\,\tilde{\boldsymbol{x}}_k$. The full state vector can be retrieved from $\boldsymbol{x}_k = \tilde{U}\tilde{\boldsymbol{x}}_k$.

### 2.4. DMD with Control

In some systems, there are input variables that drive the system dynamics (e.g. battery discharge is dependent on current input). To adjust DMD methodology to accommodate this, DMD with control seeks to estimate matrices $A$ and $B$ that approximate the nonlinear system with a linear one, written in canonical form:

$$\boldsymbol{x}_{k+1} = A\,\boldsymbol{x}_k + B\,\boldsymbol{u}_k\,.$$

In this case, in addition to $X$ and $X'$, DMD requires the construction of the input snapshot matrix $\Upsilon$:

$$\Upsilon = \begin{bmatrix} u_1(t_0) & \dots & u_1(t_i) & \dots & u_1(t_{m-1}) \\ u_2(t_0) & \dots & u_2(t_i) & \dots & u_2(t_{m-1}) \\ \dots & \dots & \dots & \dots & \dots \\ u_q(t_0) & \dots & u_q(t_i) & \dots & u_q(t_{m-1}) \end{bmatrix},$$

which leads to the state equation in matrix form:

$$X' \approx A\,X + B\,\Upsilon.$$

To solve, we can simply stack matrices $X$ and $\Upsilon$ along the state dimension, i.e., $\Omega = \left[\begin{smallmatrix} X \\ \Upsilon \end{smallmatrix}\right]$, and then run DMD by substituting $\Omega$ to the original $X$. Matrices $A$ and $B$ then come from the splitting of the left eigenvectors $U$ into two separate components (Kutz et al., 2016):

$$\begin{aligned} A &\approx X'\,V\,\Sigma^{-1}\,U_1^*\,,\\ B &\approx X'\,V\,\Sigma^{-1}\,U_2^*\,, \end{aligned}$$

where $U_1$ and $U_2$ are the components of $U$ such that $U_1 \in \mathbb{C}^{n\times r}$, $U_2 \in \mathbb{C}^{q\times r}$, and $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$.

It should be noticed that if a reduced-rank approximation of $A$, $B$ is desired (i.e., $r < \mathrm{rank}(X)$), the user should replace the full-rank matrices with the reduced rank ones and follow the steps in Section 2.3. DMD with control has been implemented in our battery case study as the discharge is driven by the input current.

## 3. Physics-Enhanced DMD: Applications and Results

### 3.1. Toy Example: DMD and Thrown Object

As a preliminary test, we applied DMD to the simple "thrown object" model from the `prog_models` python package (Teubert, Corbetta, Kulkarni, Jarvis, & Daigle, 2022). The *ThrownObject* model is a simple nonlinear model that describes the 1D dynamics of a mass thrown vertically into the air, using a standard drag equation with coefficient $C_D$ producing the resistance force. The model is written in state-space canonical form using state vector $\boldsymbol{x} = [x, \dot{x}]^\top$,

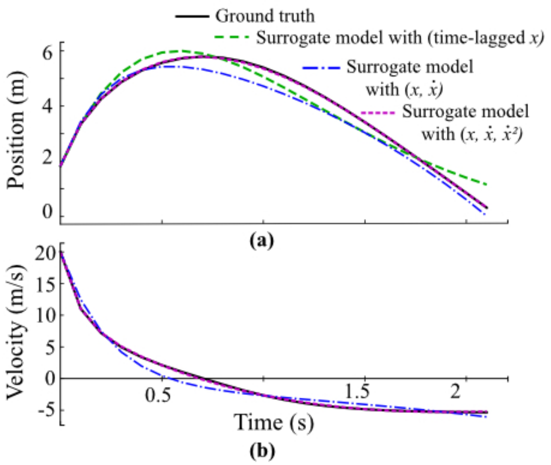

Figure 2. Prediction of a thrown object's position (a) and velocity (b) using DMD with (i) one-dimensional, lagged time series of position, (ii) position and velocity, (iii) position, velocity, and velocity-squared. Note: velocity was not predicted with the lagged time series approach, as we only introduced $x$ in the algorithm.

and measurement vector $\boldsymbol{y} = [x]$,

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \dot{x}\,, \\ \frac{\mathrm{d}\dot{x}}{\mathrm{d}t} &= -\,g - \frac{\rho C_D \mathcal{A}}{2m}|\dot{x}|\dot{x}\,, \end{aligned} \tag{5}$$

where $g$ is gravity, $m$ is the object's mass, $\mathcal{A}$ is the object reference area, and $\rho$ is the air density. The reference frame is a unit vertical axis pointing upwards normal to the ground. For this case, we ignore the effect of the small altitude changes to air density.

Beginning with a standard DMD approach, we constructed a snapshot matrix consisting of lagged time series of only the observable values, which in this case is the position of the object (same as Eq. (2) but with position instead of voltage). By using lagged-time series, one may resort to using longer lagged series to improve predictions (for example, using 10 or more previous position values in each column, i.e. $j = 10$ in Eq. (2)). To maintain a fair comparison with the physics-based method described below, we used a number of lagged steps equal to the number of state variables, i.e. 2 (for position and velocity). Next, we considered the addition of physics-based information into the snapshot matrix. To do so, we construct a snapshot matrix that includes internal states at each point in time, such that both position and velocity are included in $X$ and $X'$, giving

$$X = \begin{bmatrix} x(t_0) & \dots & x(t_i) & \dots & x(t_{m-1}) \\ \dot{x}(t_0) & \dots & \dot{x}(t_i) & \dots & \dot{x}(t_{m-1}) \end{bmatrix}\,, \tag{6}$$

and the corresponding $X'$ shifted by one time step.

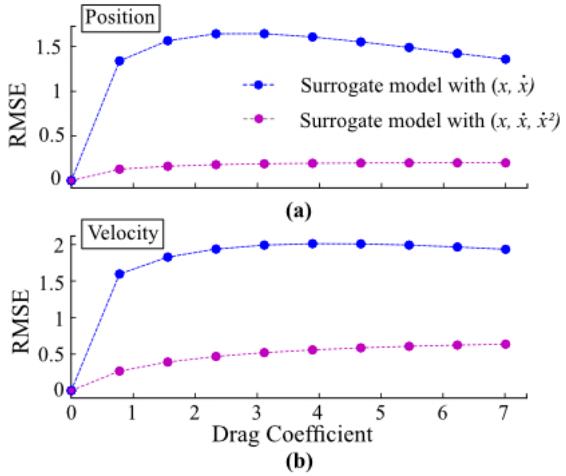

Figure 3. Root mean square error of the reconstructed position (a) and velocity (b) using DMD, as a function of the drag coefficient $C_D$. The introduction of $\dot{x}^2$ in matrices $X$, $X'$ improves predictions.

Further, we explore the configurability of a "physics-enhanced" DMD by augmenting the state vector with $\dot{x}^2$ (i.e. same as Eq. (6) with an additional row of $\dot{x}^2$ throughout time). This addition is obvious for this specific toy problem, as $\dot{x}^2$ appears in the state-space model (Eq. (5)). However, which additional terms to include may not be clear when creating surrogates of complex models.

It should be noted, nonetheless, that the results shown in Fig. 2 were obtained by introducing gravity in the matrices $X$ and $X'$ as a constant value, i.e., augmenting the state vector with $g$. By so doing, the linear case with $C_D = 0$ can be reconstructed even with no additional physics-related variables. This is strictly related to the addition of a "constant" to be estimated in linear regression, corresponding to the zero-order parameter or intercept term. Regardless of the introduction of $g$, we were unable to reconstruct the model analytically when $C_D > 0$, because we were unable to find a "closed" finite-dimensional linear approximation of it (Kutz et al., 2016).

We compared the outcomes of the three scenarios: (i) lagged time series, (ii) $x$, $\dot{x}$, and (iii) $x$, $\dot{x}$, $\dot{x}^2$, in the snapshot matrices, against a ground truth test case. The initial condition of the object was set to $x_0 = 1.83\,\mathrm{m}$ and $\dot{x}_0 = 20\,\mathrm{m/s}$, and we used the following parameters:

$$\begin{aligned} C_D &= 1.1\,, \\ g &= 9.81\,\mathrm{m/s^2}\,, \\ m &= 0.02\,\mathrm{kg}\,, \\ \mathcal{A} &= 0.01\,\mathrm{m^2}\,,\text{ and} \\ \rho &= 1.225\,\mathrm{kg/m^3}. \end{aligned}$$

After estimating matrix $A$, and setting the appropriate initial conditions, we then applied $A$ iteratively to the augmented state vector to approximate the dynamics throughout time. The results illustrate that physics-enhanced DMD performs better than a standard DMD approximation, and that the introduction of $\dot{x}^2$ allows for even more accurate predictions (Fig. 2). The significant improvement of the DMD approximation upon the addition of internal states in the snapshot matrix highlights the benefit of including physics-based information in DMD.

As a final consideration, we explored the approximation accuracy as the thrown object model became increasingly nonlinear, by sweeping through $C_D$ values from 0 (linear) up to an unreasonably high drag coefficient of 7.0. Training and testing the model for various drag coefficients, we find that the accuracy with which DMD captures the system is nearly constant, but the root mean square error obtained by including $\dot{x}^2$ in the state vector is much lower than the surrogate model with only $x$ and $\dot{x}$ (Fig. 3). As expected, when the model is linear (i.e. $C_D = 0$), DMD both with and without $\dot{x}^2$ can reconstruct the $A$ matrix up to numerical precision, so the error is zero.

### 3.2. Application of DMD to Battery Discharge Prediction

Next we applied DMD to an electrochemistry based state-of-charge model for a Li-ion battery. The model used as a high-fidelity reference was built upon first principles derived from the work (Karthikeyan, Sikha, & White, 2008), as well as empirical evidence[1], and the interested reader is referred to (Daigle & Kulkarni, 2013) where the model was first proposed. The model used here is implemented in `prog_models` (Teubert, Corbetta, Kulkarni, Jarvis, & Daigle, 2022) as `prog_models.models.BatteryElectroChemEOD`.

It is worth noting that the state-of-charge degradation phenomena operates in shorter time horizons than some other forms of degradation. Because of this, the computational benefits from surrogate models can be unnecessary. That said, the computational costs of this model are too large for some constrained environments like those present on spacecraft or small unmanned vehicles. Future work could investigate prediction for longer time horizon degradation modes, like battery aging.

The core model equations are: (i) Nernst's equation for the equilibrium potential (Eq. (7)), and (ii) Butler-Volmer equation from which the surface overpotential (Eq. (8)) derives.

$$V_{U,i} = U_{0,i} + \frac{RT}{\tilde{m}F} \ln\left(\frac{1-\tilde{x}_i}{\tilde{x}_i}\right) + V_{ni,i}\,, \tag{7}$$

$$V_{\eta,i} = \frac{RT}{F\alpha} \operatorname{arcsinh}\left(\frac{J_i}{2J_{i0}}\right)\,. \tag{8}$$

The subscript $i$ indicates the negative or positive electrode, $i = \{n, p\}$. Term $U_{0,i}$ is the reference potential, $R$ is the universal gas constant (and should not be confused with the total resistance of the model, $R_0$), $T$ is the electrode temperature in Kelvin, $\tilde{m}$ is the number of electrons transferred in the reaction ($\tilde{m} = 1$ for Li-ion batteries), $F$ is the Faraday constant, $\tilde{x}$ is the mole fraction for the Lithium-intercalated host material, and $V_{ni}$ is the activity correction term, defined for non-ideal (ni) voltage, zero in ideal conditions. In Eq. (8), $\alpha$ is a symmetry factor, $J_i$ is the current density and $J_{i0}$ is the exchange current density, all defined in (Daigle & Kulkarni, 2013). The battery output voltage $V$ can then be approximated using:

$$V = V_{U,p} - V_{U,n} - V_0 - V_{\eta,p} - V_{\eta,n}\,,$$

where $V_0$ is the nominal voltage drop defined by the required current, $i_{app}$, multiplied by the lumped internal resistance, $R_0$. Mole fractions, amount of available Li-ions, as well as concentration gradients can be retrieved from the original paper (Daigle & Kulkarni, 2013), and are omitted for the sake of brevity.

Similar to the thrown object example, the high-fidelity model provides physics-based information about the system dynamics in the form of internal system states. We leverage this known physical information about the system by building out the snapshot matrix to include the internal state information, as well as the voltage (i.e., the observable output) and the state-of-charge of the battery throughout time. State-of-charge (SOC) is defined as the progress towards the event end-of-discharge (EOD), where an SOC of 1 corresponds to a fully charged battery and 0 corresponds to a state where the event has occurred (i.e., the battery is fully discharged). SOC is highly correlated with voltage, but they are not equivalent. Thus, the snapshot matrix $X$ is of the form,

$$X = \begin{bmatrix} x_1(t_0) & x_1(t_1) & \dots \\ x_2(t_0) & x_2(t_1) & \dots \\ \dots & \dots & \dots \\ x_8(t_0) & x_8(t_1) & \dots \\ V(t_0) & V(t_1) & \dots \\ SOC(t_0) & SOC(t_1) & \dots \\ u(t_0) & u(t_1) & \dots \end{bmatrix},$$

where $x_j$ are the eight internal states of the electrochemistry battery model, $V$ is the voltage, $SOC$ is the state-of-charge, and $u$ is the current. The battery hidden state vector $\boldsymbol{x} \in \mathbb{R}^{8\times 1}$ is composed of temperature $T$, nominal voltage drop $V_0$, surface overpotential $V_{\eta,n}$, $V_{\eta,p}$, and charges for bulk $q_{b,i}$ and surface $q_{s,i}$:

$$\boldsymbol{x} = [T, V_o, V_{\eta,n}, V_{\eta,p}, q_{b,n}, q_{s,n}, q_{b,p}, q_{s,p}]^\top\,.$$

With this structure, we constructed snapshots using a collection of training data with varying loading profiles, each se-

[1] Data available at https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

quentially stacked to create one large training dataset, and solved for the linear approximation as described in Section 2.

### 3.3. Introduction of Artificial Noise to Aid Regression

One of the challenges related to the estimation of matrices $A$ and $B$ is the potential for linear dependencies among rows of $X$ and $X'$. In fact, (Kutz et al., 2016) touches on the subject when presenting DMD with control (chapter 6, page 98). Linear dependencies in the input variables prevent the correct identification of $A$ and $B$ because the solution is not unique. We observed empirically that quasi-linearly dependent rows can also prevent DMD from finding an accurate solution, even if matrices the $X$ and $X'$ have full rank.

In the battery use case presented here, some hidden states show an almost-linearly dependent behavior, even if not exact, and DMD indeed fails to generate a reliable model. To overcome the problem, we introduced random perturbations into $X$ (and therefore $X'$), which helps the stability of DMD by "breaking" the dependency among the state variables. By adding small Gaussian noise with a standard deviation of 0.01 when building the snapshot matrices, we observed more robust matrices $A$ and $B$ and a more reliable DMD model.

It is important to note that the introduction of noise into the input data should be used carefully, as the standard DMD algorithm introduces a bias in the estimation process when noise is present (Dawson, Hemati, Williams, & Rowley, 2016). Loosely, DMD can be considered a simple linear regression approach where $X$ is the independent variable and $X'$ is the dependent variable. In linear regression, noise is supposed to be present in the dependent variable, but not in the independent one. However in our case, noise is present in both $X$ and $X'$. Thus, for a more rigorous approach, one should rely on one of the more advanced DMD techniques presented in (Dawson et al., 2016) to correct for sensor noise. However, the methods in (Dawson et al., 2016) are not tailored for DMD with control, which is necessary in our battery use-case. At this stage of the research, our attempt to use methods similar to (Dawson et al., 2016) failed at reproducing a more robust linear model. Future works will investigate the use of bias-correction methods for the case studies where control is necessary.

### 3.4. Prediction Accuracy

We performed a number of blind tests for a diverse set of loading profiles (three of which are illustrated in Fig. 4). In each test, only the initial condition of the internal states, voltage, and SOC was known, as well as the expected load throughout time. The linearized model was applied iteratively to perform a multi-step look-ahead prediction, and the results were compared to the original high-fidelity electrochemistry model, from which the surrogate model was generated. The resulting physics-enhanced DMD model provides approximations to the high-fidelity model and a significant improvement over DMD with only observable values (Fig. 4a).

Notably, the approximations are accurate for long time horizons, a new feature not achievable with a classic DMD method (e.g. until SOC $\approx$ 20% compared to SOC $\approx$ 85% for the standard, observable only surrogate). However, the DMD approximation fails to capture the dynamics near end-of-discharge (EOD) (e.g., $\geq$ 2000s in Fig. 4). We find that this final stage of discharge is too nonlinear to be well approximated with our DMD model, resulting in a poor approximation of the time of final EOD. Although the sudden drop in state-of-charge and voltage are not captured, the model performs well up to a state-of-charge (SOC) of 20%. High accuracy until $\sim$20% SOC is particularly relevant because applications involving Li-ion batteries, such as UAV operations, are typically not allowed to continue below such small SOC values. Thus, using a conservative 30% minimum SOC, the DMD predictions are satisfactory. In future work, we will explore further extensions of DMD to improve this final fit.

As a final test of using DMD to predict battery voltage discharge, we adjusted our model to be representative of an aged battery by appropriately increasing the internal resistance parameter and decreasing the maximum battery capacity parameter. In doing so, the high-fidelity model describes a battery that has degraded due to aging processes (Daigle & Kulkarni, 2016). Training on similar data, we find that the DMD model is able to replicate the voltage discharge of the aged battery (Figure 4b). Thus, the physics-enhanced DMD model captures the voltage discharge dynamics of both a healthy and aged battery, and while we cannot conclude that it is also able to capture aging degradation dynamics, these findings provide positive preliminary results for future work.

### 3.5. Prognostic Performance

One utility of a surrogate model is its ability to be used in prognostics. So, as a final exploration into the benefits of a physics-enhanced DMD, we considered the use of a DMD approximation within a prognostics framework. It was expected that DMD would provide a reduction in computation time at the cost of prediction accuracy.

To evaluate the prognostic performance, we calculate the mean squared error (MSE) of predicted voltages at future time-points when compared to a ground truth (a metric of interest in many prognostic applications). For the ground truth, we use the results from the high-fidelity model with a time step of $\Delta t = 0.1$s. This is an appropriate comparison, as the goal of a surrogate model is to adequately represent the behavior of the high-fidelity model.

To evaluate the computational performance, we measure the CPU time per second of simulation (using `time.process_time` in python, average of 50 runs re-

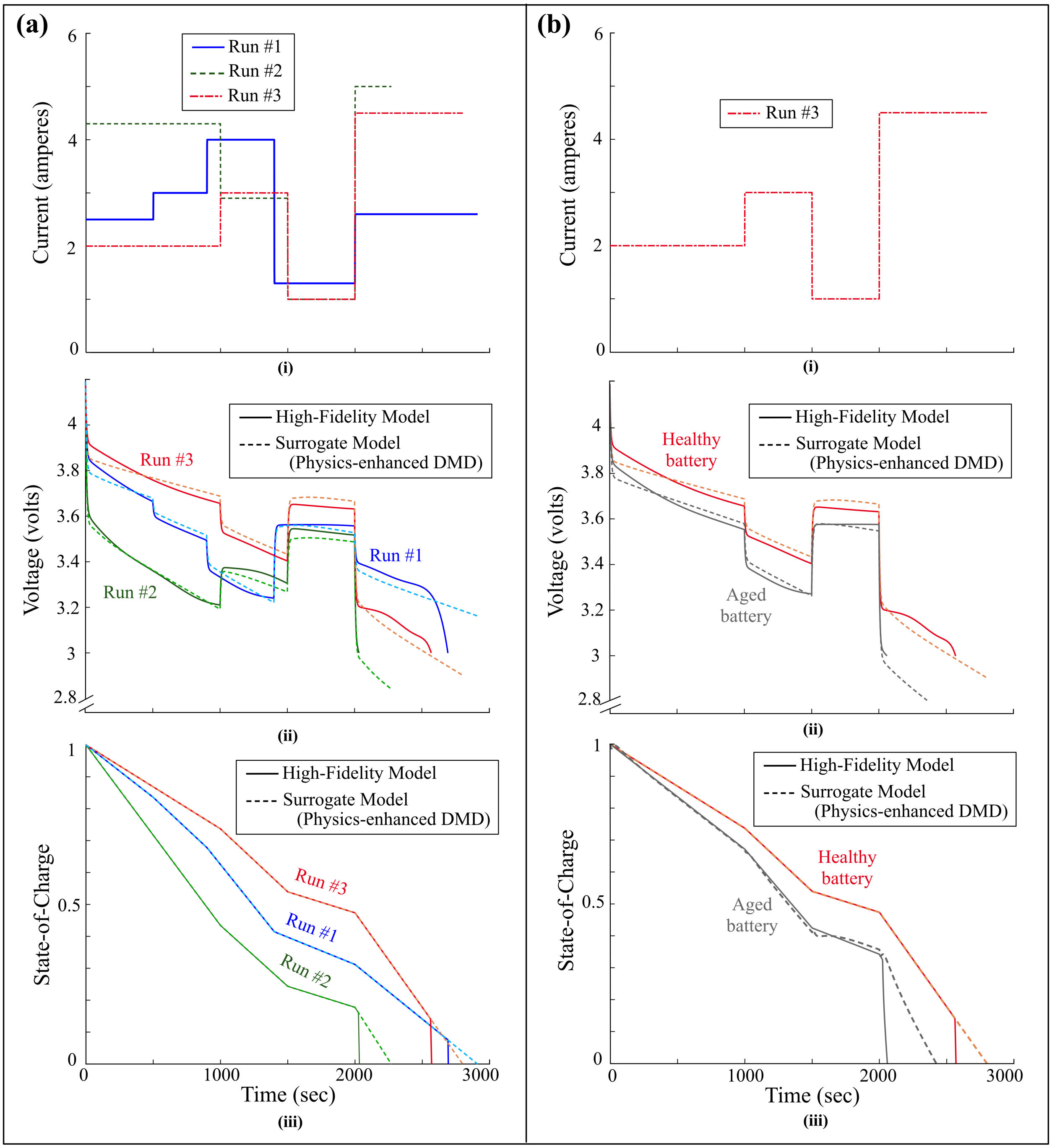

Figure 4. (a) Prediction of battery voltage (ii) and battery SOC (iii) given loading profiles (i) using the electrochemistry model (solid lines) and a physics-enhanced DMD method (dashed lines) for healthy battery. (b) Same, comparing a healthy and an aged battery.

ported in Table 1). Standardizing per second of simulation prevents an overestimation or underestimation of EOD from affecting the metric, allowing for a fair comparison of the computational efficiency of the methods. The results of this investigation are illustrated in Table 1 and Fig. 5.

Results illustrate that prognostics performance is significantly better for the high-fidelity model at small step sizes ($\Delta t$). At comparable step sizes, the surrogate model produces only modest or no performance improvements (e.g. compare $\Delta t$=5 for the two models in Table 1). However, the performance

Table 1. Prognostic and Computational Performance at Different Step Sizes

| **Model** | **$\Delta$t (s)** | **CPU-time per sim second**<br>(seconds, mean ± SD) | **Voltage MSE**<br>(volts) |
|---|---|---|---|
| High-fidelity | $\Delta t = 5$ | $(9.53 \pm 0.37) \cdot 10^{-5}$ | $4.10 \cdot 10^{-5}$ |
| High-fidelity | $\Delta t = 10$ | $(4.83 \pm 0.23) \cdot 10^{-5}$ | $7.60 \cdot 10^{-4}$ |
| High-fidelity | $\Delta t = 28$ | $(2.08 \pm 0.19) \cdot 10^{-5}$ | 18.47 |
| Physics-enhanced DMD | $\Delta t = 5$ | $(6.35 \pm 1.96) \cdot 10^{-5}$ | $1.04 \cdot 10^{-3}$ |
| Physics-enhanced DMD | $\Delta t = 10$ | $(3.26 \pm 2.03) \cdot 10^{-5}$ | $9.80 \cdot 10^{-4}$ |
| Physics-enhanced DMD | $\Delta t = 28$ | $(1.09 \pm 0.09) \cdot 10^{-5}$ | $7.26 \cdot 10^{-4}$ |

of the high-fidelity model decreases dramatically as the step size increases (Jarvis, Teubert, Okolo, & Kulkarni, 2022). At larger step sizes, the DMD model outperforms the high-fidelity model significantly (Fig. 5 and $\Delta t$=28 in Table 1).

Applications using prognostics generally define a prognostics performance requirement. For example, for a battery application with an acceptable MSE of $8 \cdot 10^{-4}$, this can only be achieved for the high-fidelity model with a $\Delta t$ of 10 seconds or less. Using the surrogate model, the same accuracy can be achieved with a $\Delta t$ of 28 seconds. For the same accuracy, the surrogate model would perform the simulation in almost $1/5$ of the time required by the equivalent higher-fidelity model.

The identification of proper $\Delta t$ is very application-dependent. Some applications have rapidly varying inputs or require a precise end-of-discharge estimate, and may not be able to use a $\Delta t$ above 10 seconds, in which case this surrogate model would only offer small performance improvement. For other applications, a $\Delta t$ above 28 seconds is acceptable.

## 4. DISCUSSION AND FUTURE WORK

This work explored the use of DMD as a surrogate modeling tool for long-term predictions in the context of battery discharge prognostics. We made use of an extended DMD approach to lift the state vector to a higher dimension in the search for a finite approximation of the Koopman operator. Instead of using basis functions or other traditional machine learning tools, we leveraged the hidden states of the higher-fidelity Li-ion battery model to construct the snapshot matrices $X$ and $X'$, thus including state variables inherently linked to the physics of the system.

The introduction of physics-related quantities enabled the creation of a successful DMD-based surrogate model. While we observed some accuracy loss when predicting voltage discharge (as expected with a surrogate model), the physics-enhanced DMD method was able to make acceptable predictions for the entire duration of the discharge process, which was the goal of out work. Other techniques such as lagged time-series (Tirunagari et al., 2017) and lifting through radial basis functions (Korda & Mezić, 2018) did not produce a robust model. While these methods and others can still approximate voltage and SOC appropriately for few time steps (i.e. small time horizon), they failed at predicting voltage discharge for many time steps ahead.

Our error and computational cost analysis (Section 3.5) shows performance in a prognostic setting, and highlights the advantage of using the DMD-based linear model for multi-step predictions. The time step size ($\Delta t$) used to propagate the linear dynamics can be increased drastically for physics-enhanced DMD when compared against the requirement of the full-order model, without reduction in performance. In specific applications, this can provide significant improvement in computational efficiency.

In the context of prognostics, the linearization of nonlinear dynamical systems also opens the door to faster uncertainty propagation methods, which are crucial for system diagnostics, remaining-useful-life predictions, and more in general PHM. Uncertainty in initial conditions, model parameters, and model form error can be propagated more easily in linear systems, while full-order models often require sampling-based methods (e.g., Monte Carlo) for a characterization of future state uncertainty, which can be computationally costly.

### 4.1. Limitations

The scope of this work is limited to substituting a fast surrogate model in place of a higher-fidelity model. By using internal state information from the full order model, we preclude the use of field data, from which internal states of the system cannot be measured, except in rare cases. DMD methods using only observables and/or data-driven extensions do not suffer from the same limitation, and they could be applied to experimental data as well.

The selection of the additional variables to include, either as hidden states from the full order dynamics or other physics-related quantities, remains an open problem. There is no

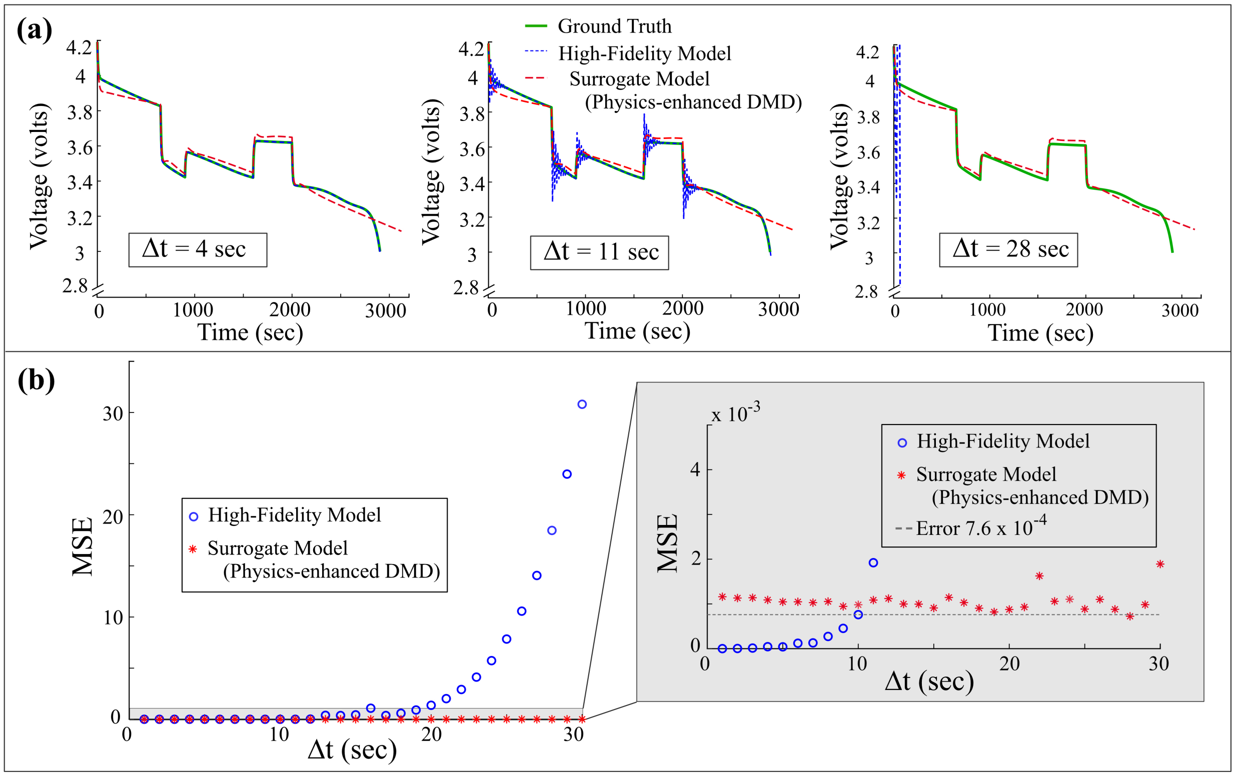

Figure 5. Simulation Benchmarking Results: (a) Voltage Curves at different $\Delta t$, and (b) MSE vs Time Step

guarantee that using the full state vector, as we did in the prognosis of battery voltage and SOC, will produce an acceptable reduced order model. At this stage of the research, we observed, empirically, that introducing all hidden states allowed us to build a battery reduced order model robust to different initial conditions and current input profiles. However, we cannot point to a rigorous approach to infer which variables should be included in $X$ and $X'$, and determining such a method is left to future work.

Additionally, it is important to reiterate that the prediction of battery discharge was chosen as a test-case with which to study the potential use of DMD in prognostics due to its relative simplicity. Therefore, long-term aging degradation for the battery was not considered. The introduction of capacity fading effects into the prediction of voltage discharge curves will likely require a DMD matrix that evolves as the system ages like, for example, a parameterized DMD model. We left this endeavor to future work.

### 4.2. Implementation in prog_models

We have implemented a generic version of this physics-enhanced DMD framework within the open-source ProgPy packages (Teubert, Corbetta, Kulkarni, Jarvis, & Daigle, 2022), allowing the surrogate method to be applied to any model created using that package. This is done using the `data_model` with the `dmd` specification. See the `generate_surrogate` example within `prog_models` for details. The resulting surrogate model can be used interchangeably for simulation or prognostics (using the `prog_algs` python package (Teubert, Corbetta, & Kulkarni, 2022)).

### 4.3. Future Work

We plan to explore further extensions to the physics-enhanced DMD method and apply it to a wider range of applications. In the context of battery discharge prediction, we hope to improve the DMD approximation to more accurately capture behavior near end of discharge. As one approach, we plan to explore the potential of fitting an entire voltage curve by stitching together linear approximations made from DMD matrices trained on different sections of the training data.

As discussed in subsection 4.1, the introduction of aging effects into the DMD matrix may enable the use of DMD for a number of additional prognostic scenarios, including prediction of remaining-useful-life for Li-ion batteries, like the one used for this work.

Additionally, it will be worthwhile to further explore the configurability of a DMD approximation by including different subsets of the known physical information. A rigorous method for determining what physical information to include would lead to a more robust physics-enhanced DMD technique.

Finally, we intend to explore alternative surrogate model generation methods, such as those emerging from the physics-informed machine learning communities, and compare with the benefits of the current DMD approach.

## ACKNOWLEDGMENT

This work was supported by (1) the System-Wide Safety (SWS) project under the Airspace Operations and Safety Program (AOSP), and (2) the Data and Reasoning Fabric (DRF) project under the Transformative Aeronautics Concepts Program (TCAP) Convergent Aeronautics Solutions (CAS) Project, both within the NASA Aeronautics Research Mission Directorate (ARMD).

## REFERENCES

Daigle, M., & Kulkarni, C. S. (2013). Electrochemistry-based battery modeling for prognostics. In *Annual conference of the phm society* (Vol. 5).

Daigle, M., & Kulkarni, C. S. (2016). End-of-discharge and end-of-life prediction in lithium-ion batteries with electrochemistry-based aging models. In *Aiaa infotech@ aerospace* (p. 2132).

Dawson, S., Hemati, M. S., Williams, M. O., & Rowley, C. W. (2016). Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, *57*(3), 1–19.

Jarvis, K., Teubert, C., Okolo, W., & Kulkarni, C. (2022). Improving computational efficiency of prognostics algorithms in resource-constrained settings. In *2022 ieee aerospace conference.*

Karthikeyan, D. K., Sikha, G., & White, R. E. (2008). Thermodynamic model development for lithium intercalation electrodes. *Journal of Power Sources*, *185*(2), 1398–1407.

Koopman, B. O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the United States National Academy of Sciences*, *17*(5), 315.

Korda, M., & Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, *93*, 149–160.

Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, *35*(4), 551–566.

Teubert, C., Corbetta, M., & Kulkarni, C. (2022, May). *Prognostics algorithm python package.* Retrieved from `https://github.com/nasa/prog_algs`

Teubert, C., Corbetta, M., Kulkarni, C., Jarvis, K., & Daigle, M. (2022, May). *Prognostics models python package.* Retrieved from `https://github.com/nasa/prog_models`

Tirunagari, S., Kouchaki, S., Poh, N., Bober, M., & Windridge, D. (2017). Dynamic mode decomposition for univariate time series: analysing trends & forecasting. *hal- 01463744*. Retrieved from `https://hal.archives-ouvertes.fr/hal-01463744`

Williams, M. O., Kevrekidis, I. G., & Rowley, C. W. (2015). A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, *25*(6), 1307–1346.

## BIOGRAPHIES

**Katelyn Jarvis** is an applied mathematician and research engineer at NASA Ames Research Center. Katy received her B.S. in Mathematics and M.S. and Ph.D. in Applied Mathematics from the University of California, Davis. At NASA, Katy is a researcher within the System-Wide-Safety and Data and Reasoning Fabric projects, as well as a member of the Diagnostics and Prognostics group. Her research interests include development of novel algorithmic approaches to prognostics, improvement of computational efficiency of these methods, and application of prognostics in human-health related fields.

**Matteo Corbetta** is a Research Engineer with KBR Inc. at NASA Ames Research Center, CA. He is a member of the Diagnostics & Prognostics Group at Ames. He works at the intersection of physics-based models, machine learning and physics-informed learning, and uncertainty quantification methods for diagnostics and prognostics of aerospace systems. He is a member of the Editorial Board of the PHM Society.

**Christopher Teubert** is the Diagnostics and Prognostics group lead at NASA Ames Research Center. Christopher received his B.S. in Aerospace Engineering from Iowa State University and his M.S. in Computer Science and Engineering from Santa Clara University. At NASA, Christopher performs research in software architectures for prognostics, is the lead of NASA's Prognostics CoE, and is the lead developer for the Prognostics Python Packages (prog_models, prog_algs, and prog_server).

**Stefan Schuet** received the B.S. and M.S. degrees in electrical engineering from Santa Clara University, Santa Clara, CA, in 2001 and 2004, respectively. He joined NASA Ames Research Center in 2001, and currently leads research that combines applied-physics with machine learning and optimization to address challenging aerospace engineering problems. Mr. Schuet holds a U.S. patent and several distinguished awards including the 2008 NASA Government Invention of the Year Award, the 2009 R&D Magazine Top 100 Award, and the 2018 NASA Exceptional Engineering Achievement Medal.