SAND2020-7833C

Learning Compact Physics-Aware Photocurrent Models Using Dynamic Mode Decomposition

Joshua Hanson

Sandia National Laboratories

August 3, 2020

Motivation

- Radiation-induced photocurrent in semiconductor devices has traditionally been simulated using complex physics-based models, which are accurate, but often computationally expensive to solve numerically.
- We want to develop compact photocurrent models that are efficient for use in large-scale circuit simulators, but remain faithful to the underlying physics.
- We approach this problem by using Dynamic Mode Decomposition (DMD) to model the internal device dynamics derived from physics-based simulations. We use the simulated state as training data, instead of raw experimental measurements.
- This allows us to reproduce the accuracy of the physics-based models, but with the run-time efficiency of machine learning models.

Brief intro to Koopman operator theory

- Consider a discrete-time dynamical system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k)$$
$$\mathbf{x}_k \in M \hookrightarrow \mathbb{R}^n$$
$$h : M \to \mathbb{R}$$

- Learning the transition map f can be difficult for nonlinear dynamics.
- Idea: "lift" the dynamics from the manifold M to a Hilbert space $\mathcal{O}(M)$ of observables. Want the lifted dynamics to be linear.
- Definition: The *Koopman operator* is the infinite-dimensional linear map given by

$$\mathcal{K} : \mathcal{O}(M) \to \mathcal{O}(M)$$
$$h \mapsto h \circ f$$

Brief intro to Koopman operator theory

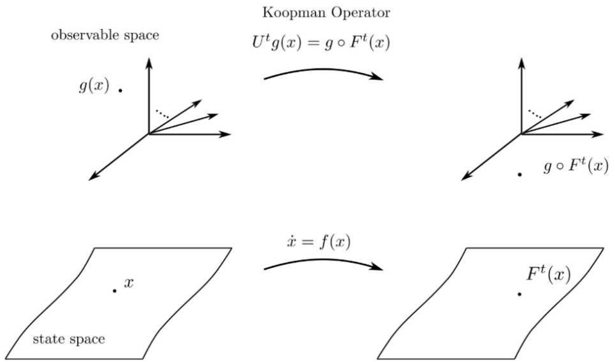

Figure 0.1: Koopman viewpoint lifts the dynamics from state space to the observable space, where the dynamics is linear but infinite dimensional.

Figure: Hassan Arbabi, Intro. to Koopman operator theory of dynamical systems, 2018

Approximating the Koopman operator

- Under certain assumptions, $\mathcal{K}$ is a compact operator. Let (g_i, λ_i) represent the eigenfunction/eigenvalue pairs (i.e. $\mathcal{K} g_i = \lambda_i g_i$).
- We can approximate $\mathcal{K}$ by restricting to a finite-dimensional observable subspace

$$H_N := \text{span}\{h_1, \dots, h_N\} \subset \mathcal{O}(M)$$

- Let $h(\mathbf{x}) := \sum_{i=1}^{N} c_i h_i(\mathbf{x})$. Then

$$h(\mathbf{x}_{k+1}) = \mathcal{K}^{k+1} h(\mathbf{x}_0) \approx \sum_{i=1}^{N} c_i \lambda_i^{k+1} h_i(\mathbf{x}_0)$$

- We can recover $\mathbf{x}$ by applying h^{-1}. The approximation error will depend on how "expressive" H_N is.

Dynamic mode decomposition (DMD)

- Suppose we have samples of the state vector

$$X = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_{m-1} \\ | & & | \end{bmatrix}; \quad X' = \begin{bmatrix} | & & | \\ \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & & | \end{bmatrix}$$

- If the dynamics are linear, we have $X' = AX$ for some $A \in \mathbb{R}^{n \times n}$.
- We can construct a data-driven model by approximating the transition map via $A \approx \bar{A} := X'X^\dagger$ ($\dagger$ = Moore-Penrose pseudoinverse).
- The pseudoinverse is best approximated (in spectral norm) via truncated SVD:

$$X = U\Sigma V^\mathsf{T} = \begin{bmatrix} \tilde{U} & \tilde{U}_{\text{trun}} \end{bmatrix} \begin{bmatrix} \tilde{\Sigma} & 0 \\ 0 & \tilde{\Sigma}_{\text{trun}} \end{bmatrix} \begin{bmatrix} \tilde{V}^\mathsf{T} \\ \tilde{V}_{\text{trun}}^\mathsf{T} \end{bmatrix} \approx \tilde{U}\tilde{\Sigma}\tilde{V}^\mathsf{T}$$

$$X^\dagger \approx \tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^\mathsf{T}$$

$U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m-1}$, $V^\mathsf{T} \in \mathbb{R}^{m-1 \times m-1}$, $\tilde{U} \in \mathbb{R}^{n \times p}$, $\tilde{\Sigma} \in \mathbb{R}^{p \times p}$, $\tilde{V}^\mathsf{T} \in \mathbb{R}^{p \times m-1}$

Dynamic mode decomposition (DMD)

- If the dynamics are nonlinear, can use $\mathcal{K}$ to lift to a high-dimensional approximate linear model, then apply DMD as before:

$$\mathbf{h}(X) = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_1(\mathbf{x}_{m-1}) \\ | & & | \\ h_N(\mathbf{x}_1) & \cdots & h_N(\mathbf{x}_{m-1}) \end{bmatrix}; \quad \mathbf{h}(X') = \begin{bmatrix} h_1(\mathbf{x}_2) & \cdots & h_1(\mathbf{x}_m) \\ | & & | \\ h_N(\mathbf{x}_2) & \cdots & h_N(\mathbf{x}_m) \end{bmatrix}$$

$$\mathbf{h}(X') = \mathcal{K}\mathbf{h}(X)$$

$$\mathcal{K} \approx \bar{\mathcal{K}} := \mathbf{h}(X')\mathbf{h}(X)^{\dagger} \approx \mathbf{h}(X')\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^{\mathsf{T}}$$

- To reduce the model order, transform into normal dynamic mode coordinates:

$$\mathbf{h}(\mathbf{x}_{k+1}) = \bar{\mathcal{K}}\mathbf{h}(\mathbf{x}_k), \quad \bar{K} \in \mathbb{R}^{N \times N}$$

$$\tilde{\mathbf{x}} \leftarrow \tilde{U}^{\mathsf{T}}\mathbf{h}(\mathbf{x})$$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathcal{K}}\tilde{\mathbf{x}}_k, \quad \tilde{K} = \tilde{U}^{\mathsf{T}}\bar{K}\tilde{U} \in \mathbb{R}^{r \times r}$$

Dynamic mode decomposition with control (DMDc)

- If the state space model has an exogeneous input, we can accomodate it by augmenting the sample matrix:

$$X = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_{m-1} \\ | & & | \end{bmatrix}; \quad X' = \begin{bmatrix} | & & | \\ \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & & | \end{bmatrix}; \quad G = \begin{bmatrix} | & & | \\ \mathbf{g}_1 & \cdots & \mathbf{g}_{m-1} \\ | & & | \end{bmatrix}$$

$$X' = AX + BG = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ G \end{bmatrix} =: \begin{bmatrix} A & B \end{bmatrix} S$$

$$\begin{bmatrix} A & B \end{bmatrix} \approx \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix} := X' S^\dagger$$

- Let the truncated SVD of S and X' be denoted by

$$S^\dagger \approx \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^\mathsf{T}, \; X' \approx \hat{U} \hat{\Sigma} \hat{V}^\mathsf{T}$$

$$\tilde{U} = \begin{bmatrix} \tilde{U}_1^\mathsf{T} & \tilde{U}_2^\mathsf{T} \end{bmatrix}^\mathsf{T}, \; \tilde{U}_1 \in \mathbb{R}^{n \times p}, \; \tilde{U}_2 \in \mathbb{R}^{n \times p}$$

Dynamic mode decomposition with control (DMDc)

- The transition and input maps of the reduced-order model can now be approximated by

$$A \approx \bar{A} \approx X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U_1}^\mathsf{T} \in \mathbb{R}^{n\times n}$$
$$B \approx \bar{B} \approx X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U_2}^\mathsf{T} \in \mathbb{R}^{n\times n}$$

$$\tilde{A} := \hat{U}^\mathsf{T}\bar{A}\hat{U} = \hat{U}^\mathsf{T}X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U_1}^\mathsf{T}\hat{U} \in \mathbb{R}^{r\times r}$$
$$\tilde{B} := \hat{U}^\mathsf{T}\bar{B} \quad = \hat{U}^\mathsf{T}X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U_2}^\mathsf{T} \quad \in \mathbb{R}^{r\times n}$$

Application: photocurrent in PN diode

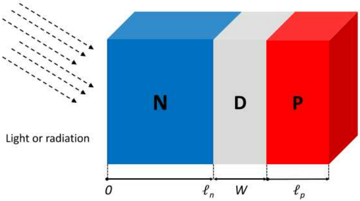

Figure: PN diode

- The delayed photocurrent in n region can be modeled by the ambipolar diffusion equation

$$\frac{\partial u}{\partial t} = D_a \nabla^2 u - \mu_a \mathbf{E} \cdot \nabla u - \frac{1}{\tau_a} u + g$$

- Can discretize to a system of ODEs by partitioning the region into a finite number of *elements* and approximating the solution by $u(x,t) \approx \sum_{i=1}^{N} c_i(t)\phi_i(x)$ where ϕ_i, are called *shape functions*.

Physics-based model development outline

1. Experimental measurements $g_{\text{train}} \longrightarrow J_{\text{left}}, J_{\text{right}}$
2. Discretize physics-based model (FEM)

$$\frac{\partial u}{\partial t} = D_a \nabla^2 u - \mu_a \mathbf{E} \cdot \nabla u - \frac{1}{\tau_a} u + g$$
$$\longrightarrow M\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{g}$$

3. Numerically compute interal state (i.e. excess carrier density)

$$M\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{g}_{\text{train}}$$
$$\longrightarrow X,\ X',\ G$$

4. Train reduced-order model (DMD)

$$X,\ X',\ G$$
$$\longrightarrow \tilde{\mathbf{x}}_{k+1} = \tilde{A}\tilde{\mathbf{x}}_k + \tilde{B}\mathbf{g}$$

Simulation details

- Excess carrier density was simulated using
  - 1024 mesh elements with linear shape functions (triangle "hats" centered at each node).
  - 2000 time steps, corresponding to a sampling interval of 1.25 ns and horizon of 2.5 μs.
- Independent models must be trained for each electric field strength.
- Simulated carrier density and flux can be reconstructed by multiplying the state of the of DMD model by the finite element shape functions or their gradients.

Model training

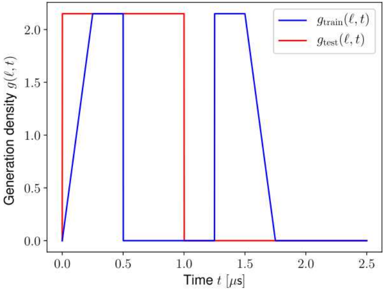

Figure: Training input and test input functions (uniform in space).

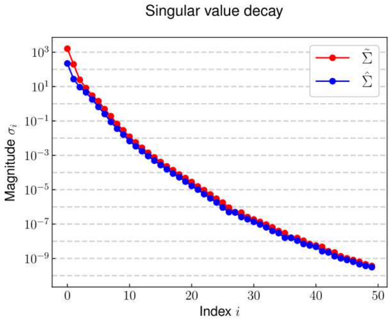

Figure: Singular values for decomposed sample matrices S (red) and X' (blue).

Results

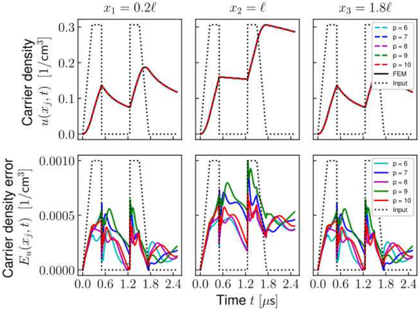

Figure: Simulated excess carrier density $u(x,t)$ due to g_{train} with no E field.

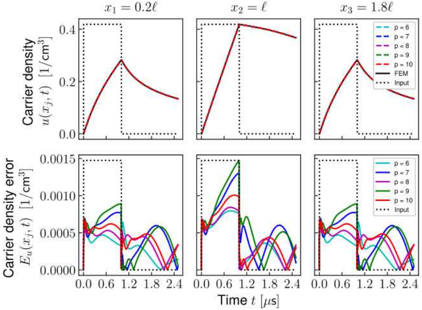

Figure: Simulated excess carrier density $u(x,t)$ due to g_{test} with no E field.

Results

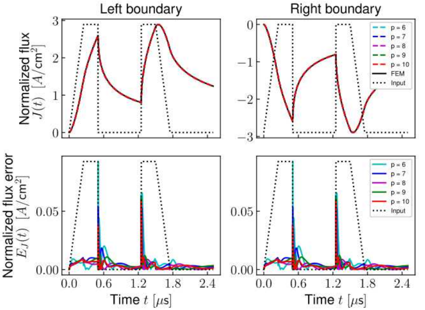

Figure: Simulated carrier flux $J(t)$ at boundaries due to g_{train} with no E field.

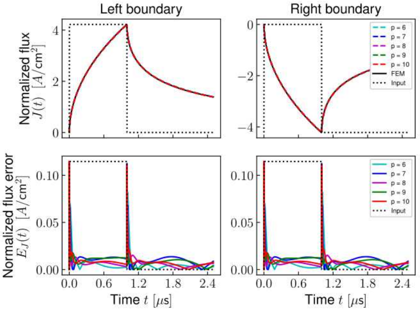

Figure: Simulated carrier flux $J(t)$ at boundaries due to g_{test} with no E field.

Results

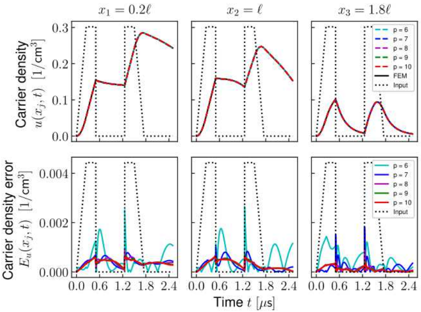

Figure: Simulated excess carrier density $u(x,t)$ due to g_{train} with E field.

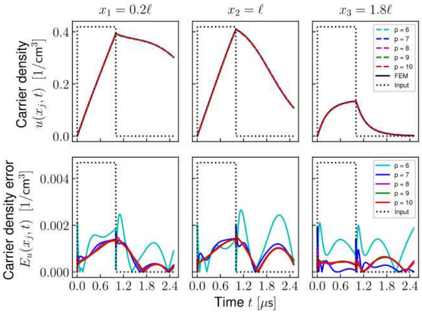

Figure: Simulated excess carrier density $u(x,t)$ due to g_{test} with E field.

Results

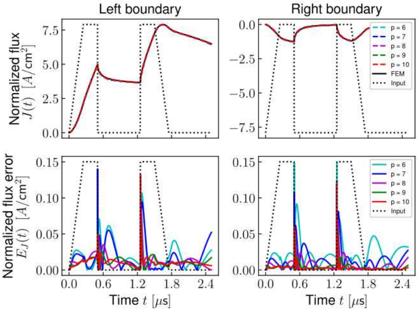

Figure: Simulated carrier flux $J(t)$ at boundaries due to g_{train} with E field.

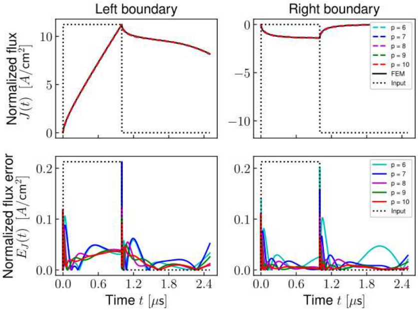

Figure: Simulated carrier flux $J(t)$ at boundaries due to g_{test} with E field.

Future work

- We can accomodate variable electric field strengths by training a *bilinear model*. This can still be done using DMD, but requires some care in designing the training input.
- The same modeling scheme can be applied to more accurate nonlinear physics-based models, such as the drift-diffusion equations, with an appropriate dictionary of observables.
- DMD generates a discrete-time model, but this should be converted to a continuous-time model for use in ODE solvers and circuit simulators (transform poles via complex logarithmic map).