Order reduction for a signaling pathway model of neuronal synaptic plasticity

Mikko Lehtimäki, Lassi Paunonen, Seppo Pohjolainen, Marja-Leena Linne

Tampere University of Technology, Finland

July 13th, 2017

Main Objectives

- Study a mathematical model of plasticity of a single biological neuron in a rodent brain
- Plasticity describes the cell's ability to "learn"
- Model introduced by Kim, Hawes, Gillani, Wallace & Blackwell in 2013
- The model is comprehensive, but too complex for network simulations consisting of large numbers of neurons
- Motivation: Demand for increase of computational efficiency

Main Objectives

- Study a mathematical model of plasticity of a single biological neuron in a rodent brain
- Plasticity describes the cell's ability to "learn"
- Model introduced by Kim, Hawes, Gillani, Wallace & Blackwell in 2013
- The model is comprehensive, but too complex for network simulations consisting of large numbers of neurons
- Motivation: Demand for increase of computational efficiency

Main contributions

- **Model reduction** for the model by Kim *et. al.*
- Introduction of new model reduction techniques in computational neuroscience.

Challenges:

- The model is nonlinear and has time-dependent dynamics.

The Plasticity Model

The model:

$$\dot{x}(t) = (A_0 + A_1 Ca(t) + A_2 Ca(t)^2 + A_3 Glu(t))x(t) + F(x(t)) + B \cdot Glu(t)$$

Main characteristics:

- $x(t) \in \mathbb{R}^n$ with $n = 44$.
- Has two external stimuli, $Ca(t)$ and $Glu(t)$
- $Ca(t)$ and $Glu(t)$ appear bilinearly (and $Glu(t)$ also linearly)
- The nonlinearity $F(x(t))$ is quadratic
- $x(t)$ includes 5 biologically interesting quantities ($=$ outputs)

The Plasticity Model

The model is mathematically challenging due to bilinearity.

In this study: Model reduction is completed for fixed and biologically motivated stimulus functions $Ca(t)$ and $Glu(t)$ of form

$$f(t) = \begin{cases} 0 & 0 \leq t \leq 5 \\ \text{sinusoidal oscillation} & 5 < t < 10 \\ 0 & t \geq 10 \end{cases}$$

The model has the form

$$\dot{x}(t) = A(t)x(t) + F(x(t)) + B \cdot Glu(t)$$

Specific aims:

- Reduce dimension of the system and the computational complexity of simulations.
- Compare the outputs (biologically interesting quantities) for the original and reduced systems.

Reduction Methods — POD and DEIM

In the model reduction, we combine two reduction techniques:

Proper Orthogonal Decomposition (POD)

and

Discrete Empirical Interpolation Method (DEIM)

Proper Orthogonal Decomposition (POD)

- A **well-known** and **widely used** method for model reduction ODEs, PDEs, and dynamical systems.
- Based on the procedure
  - Simulate the full-order system
  - Choose "snapshots" $S = [x(t_1), \ldots, x(t_N)]$
  - Form the SVD of S, and project the system onto the space of the k largest singular values
- Results in a reduced system capturing the dominant dynamics
- Computational burden reduced significantly if $k \ll n = 44$

Discrete Empirical Interpolation Method (DEIM)

POD results in a reduced order model

$$\dot{x}_k(t) = V_k^* A(t) V_k x_k(t) + V_k^* F(V_k x_k(t)) + V_k^* B \cdot Glu(t).$$

The main computational complexity results from evaluating

$$F(V_k x_k(t)) \in \mathbb{R}^n$$

($F(\cdot)$ has $n = 44$ component functions) at each time-step.

DEIM is a method developed to reduce this complexity.

Discrete Empirical Interpolation Method (DEIM)

Main steps for simplifying $V_k^* F(V_k x_k(t))$:

- The idea is to choose a subset $\{j_1, \ldots, j_m\}$ of indices $\{1, \ldots, n\}$, and choose the component functions of $F(\cdot)$ that are **most relevant** for the dynamics.
- Can replace $F(V_k x_k(t))$ with $F_m(V_k x_k(t))$ where F_m has only $m \ll n = 44$ components
- The indices $\{j_1, \ldots, j_m\}$ are chosen using an algorithm [Chaturantabut & Sorensen, 2011], and the "snapshots" in the POD reduction.

Discrete Empirical Interpolation Method (DEIM)

Benefits of simplifying $V_k^* F(V_k x_k(t))$:

- Results in a significant improvement in reduction of computation times.
- No additional simulations needed, since the "snapshots" can be collected at the same time as the simulation for POD is completed.

Summary of the Reduction Methods

To summarize:

(1) POD is used to reduce the dimension of the system

(2) DEIM is used to improve the bottle-neck of POD that results from needing to compute the nonlinear term $F(V_k x_k(t))$

The combination results in significant computational savings:

- Demonstrated in the literature especially for reduction of numerical approximations of nonlinear PDEs.
- Here we apply it in computational neuroscience.

Simulation Results

Setup:

- Consider changes in computational performance and error of the solution of

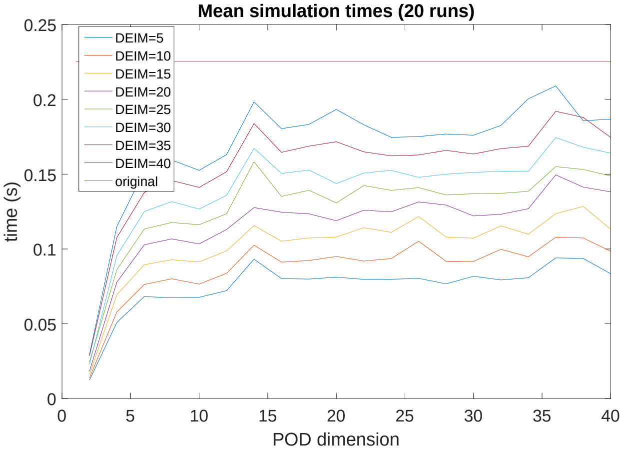
$$\dot{x}(t) = A(t)x(t) + F(x(t)) + B \cdot Glu(t)$$

  and especially the 5 chosen biologically interesting variables.
- Compare simulation times for different sizes of the POD and DEIM approximations

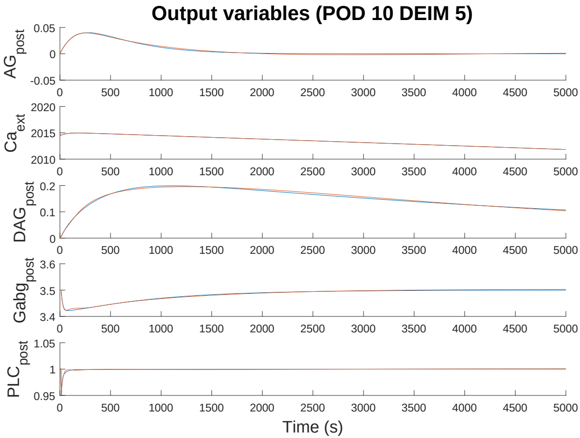
The stimuli $Ca(t)$ and $Glu(t)$ are chosen to have form

$$f(t) = \begin{cases} 0 & 0 \leq t \leq 5 \\ \text{sinusoidal oscillation} & 5 < t < 10 \\ 0 & t \geq 10 \end{cases}$$

Simulation times vs. POD and DEIM dimensions

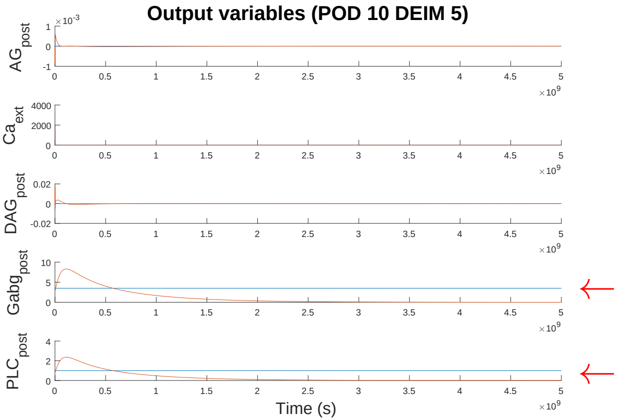


After $\dim_{POD} = 15$, POD dimension makes little difference.

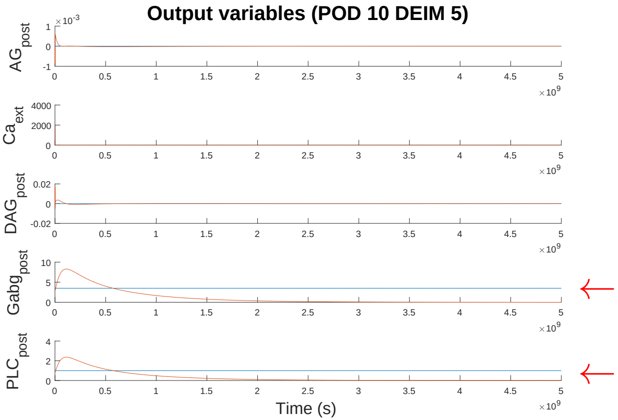


- Good accuracy with significant improvement in computational efficiency ($\approx 35\%$ of the original simulation time).

- Crucial restriction: Model not accurate for time-intervals longer than the simulation used in the POD approximation.



- Crucial restriction: Model not accurate for time-intervals longer than the simulation used in the POD approximation.



- Can be compensated by increasing the sizes of the POD and DEIM approximations $\Rightarrow$ Loss of computational savings.

Conclusions: Drawbacks and Restrictions

- The performance of the reduced model can be guarangeed for the stimulus signals $Ca(t)$ and $Glu(t)$ used in forming the POD+DEIM reduction
  - Can be improved: POD can be formed using simulations for several signals, and reduced model works for combinations of these basic signals.
  - $\Rightarrow$ Approximation works for a larger class of stimuli
- Performance can only be guaranteed on the time-interval used in forming of the POD.
  - Can be improved by increasing the size of the approximation, but cancels computational savings.
  - $\Rightarrow$ important to guarantee sufficiently long time interval for POD approximation.

Conclusions

In this presentation:

- Model reduction techniques for improving computational efficiency of a biological neuron model.
- Model has quadratic nonlinearity and bilinear features.
- Reduction methods combine
  - **Proper Orthogonal Decomposition (POD)** and
  - **Discrete Empirical Interpolation Method (DEIM)**
- Illustration of performance gains with simulations.

Thank You!