Model reduction of nonlinear feedback systems by data-informed balancing

With application to RNNs and networked systems

Anastasia Bizyaeva, Steven Brunton, Nathan Kutz

UNIVERSITY of WASHINGTON

AI INSTITUTE IN DYNAMIC SYSTEMS

My research in one slide

Complex systems in nature are **dynamic** and **nonlinear.** *E.g. neuron ensembles, biological swarms*

I try to understand and design bio-inspired autonomy and artificial intelligence using tools from **nonlinear dynamical systems** and **control theory**.

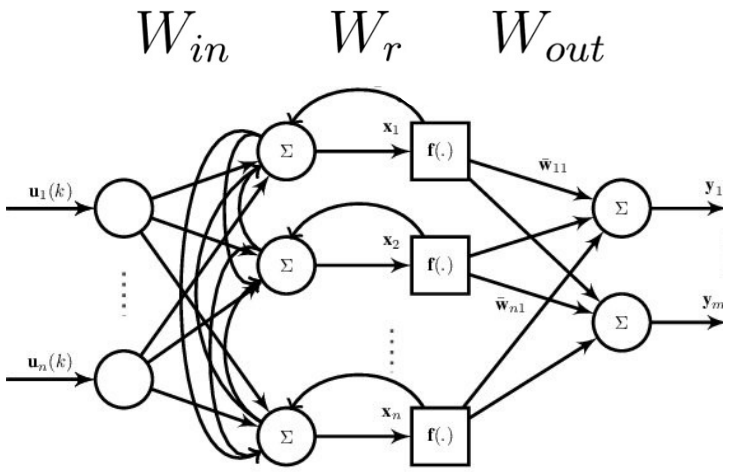
Some themes in my work:

❑ **Biologically inspired autonomy:** what mathematical principles explain adaptability and robustness of behaviors in nature? How can we design engineered systems that have similar properties?
E.g. bio-inspired collective decision-making, swarm intelligence, neuromorphic control strategies

❑ **Dynamic neural networks:** can we understand how neural networks learn and embed information?
E.g. echo state networks, spiking neural networks

A brief history of reservoir computing

Recurrent neural networks



Single hidden layer RNN:

$$r(t+1) = \tanh(W_r r(t) + W_{in} u(t))$$

$$y(t) = W_{out} r(t)$$

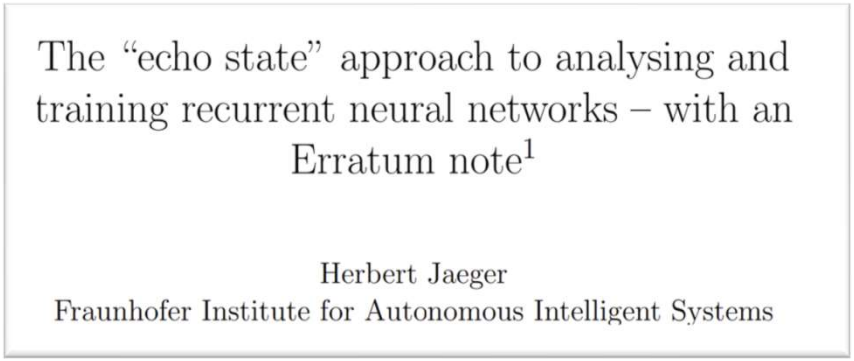
RNNs often used with time series data (e.g. for forecasting)

"Conventional" training: update W_r, W_{in}, W_{out} using gradient descent + backpropagation through time

- ❑ Computationally expensive (prohibitively so in the early 2000s!)
- ❑ Finnicky, often runs into issues (e.g. vanishing gradients)

Reservoir computing: an alternative approach to RNN training

Echo State Networks (2001)



The "echo state" approach to analysing and training recurrent neural networks – with an Erratum note[1]

Herbert Jaeger
Fraunhofer Institute for Autonomous Intelligent Systems

Liquid state machines (2002)

> Neural Comput. 2002 Nov;14(11):2531-60. doi: 10.1162/089976602760407955.

Real-time computing without stable states: a new framework for neural computation based on perturbations

Wolfgang Maass [1], Thomas Natschläger, Henry Markram

Affiliations + expand

PMID: 12433288 DOI: 10.1162/089976602760407955

Abstract

A key challenge for neural modeling is to explain how a continuous stream of multimodal input from a rapidly changing environment can be processed by stereotypical recurrent circuits of integrate-and-fire neurons in real time. We propose a new computational model for real-time computing on time-varying input that provides an alternative to paradigms based on Turing machines or attractor neural networks. It does not require a task-dependent construction of neural circuits. Instead, it is based on principles of high-dimensional dynamical systems in combination with statistical learning theory and can be implemented on generic evolved or found recurrent circuitry. It is shown that the inherent transient dynamics of the high-dimensional dynamical system formed by a sufficiently large and heterogeneous neural circuit may serve as universal analog fading memory. Readout neurons can

$$r(t+1) = \tanh(W_r r(t) + W_{in} u(t))$$
$$y(t) = W_{out} r(t)$$

Training echo state networks

Goal: forecast time series $X = [x(0), x(1), ..., x(N)]$

Step 1 (training): initialize W_r, W_{in} randomly, force RNN with $u(t) = x(t)$
collect reservoir response time series $R = [r(1), r(2), \ldots, r(N+1)]$
Look for ouput weights that map response to input $Y = W_{out} R = X$

Step 2 (forecasting): $r(t+1) = \tanh((W_r + W_{in} W_{out}) r(t)), t = N+1, \ldots$

[1] Mantas Lukoševičius and Herbert Jaeger, Reservoir computing approaches to recurrent neural network training, *Computer science review*, 2009

$$r(t+1) = \tanh(W_r r(t) + W_{in} u(t))$$
$$y(t) = W_{out} r(t)$$

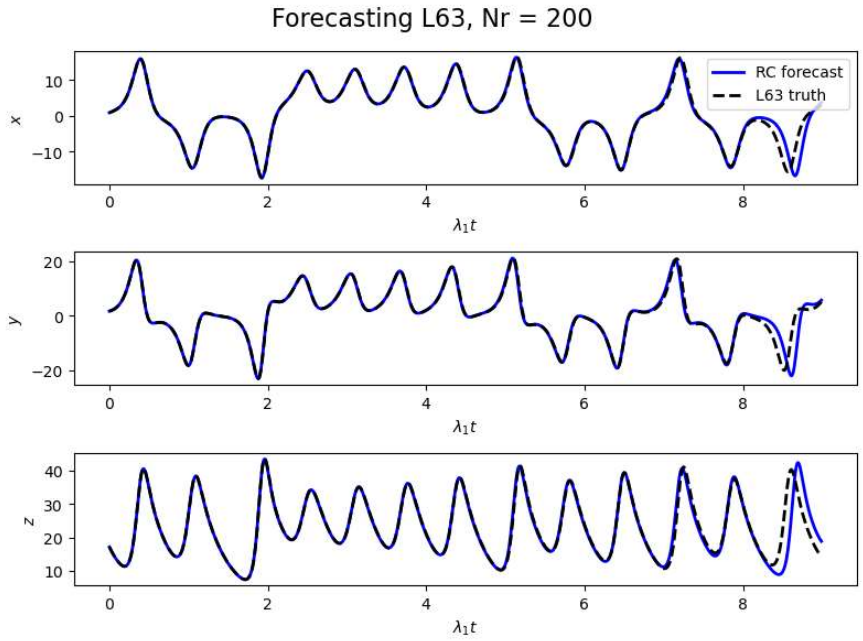
Training echo state networks

Advantages of reservoir computing approach to RNN training:

- efficiency (only W_r trained, no gradients or BPTT)
- easily implemented online (e.g. with recursive least squares algorithms)
- biologically plausible! Many well-grounded connections to architectural and dynamical properties of mammalian brains [1]

[1] Mantas Lukoševičius and Herbert Jaeger, Reservoir computing approaches to recurrent neural network training, *Computer science review*, 2009

ESNs can produce long forecasts of complex dynamics!



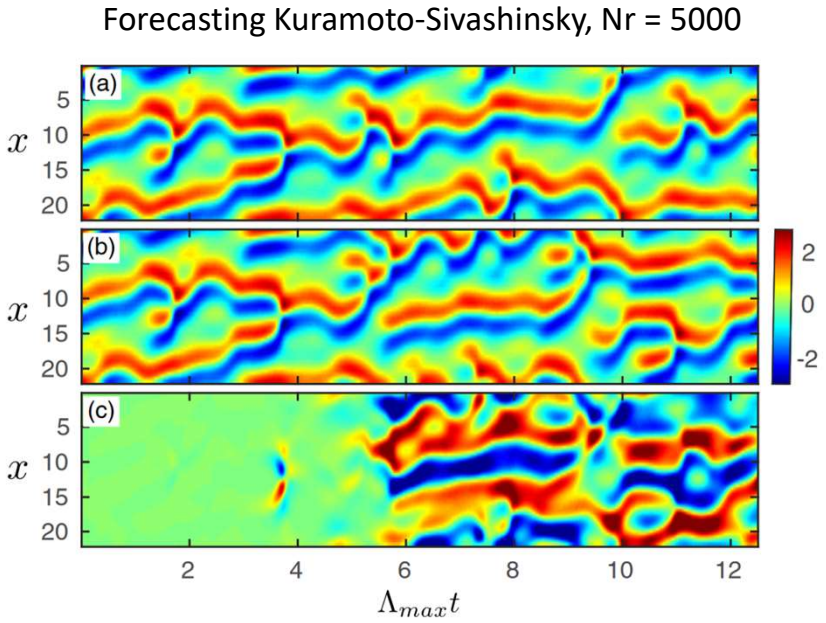


Figure from Pathak J, Hunt B, Girvan M, Lu Z, Ott E. *Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach*. PRL 2018

What does any of this have to do with model reduction?

Initialization of ESNs is still not understood theoretically, ad hoc or optimization-based hyperparameter tuning is typically required

$$r(t+1) = -\lambda r(t) + (1-\alpha)\tanh(W_r r(t) + W_{in} u(t) + b)$$

Recent literature suggests that ESNs **embed** the dynamics they are trained on [1,2]

Systems-theoretic model reduction techniques can help us decipher **how** ESNs embed information and suggest a **constructive** approach to ESN initialization.

[1] Hart A, Hook J, Dawes J. Embedding and approximation theorems for echo state networks. *Neural Networks*. 2020.
[2] Duan XY, Ying X, Leng SY, Kurths J, Lin W, Ma HF. Embedding theory of reservoir computing and reducing reservoir network using time delays. *Physical Review Research*. 2023

Feedback systems and model reduction

Model reduction (Petrov-Galerkin)

Consider a high-dimensional full-order model (FOM) of the form

$$\dot{x} = f(x), \quad x(0) = x_0, \quad y = g(x), \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m$$

Goal: find a projection $P = \Phi\Psi^T \in \mathbb{R}^{n \times n}$ that defines a reduced set of coordinates $x_r = \Psi^T x \in \mathbb{R}^r, \quad r < n$ and a reduced-order model (ROM)

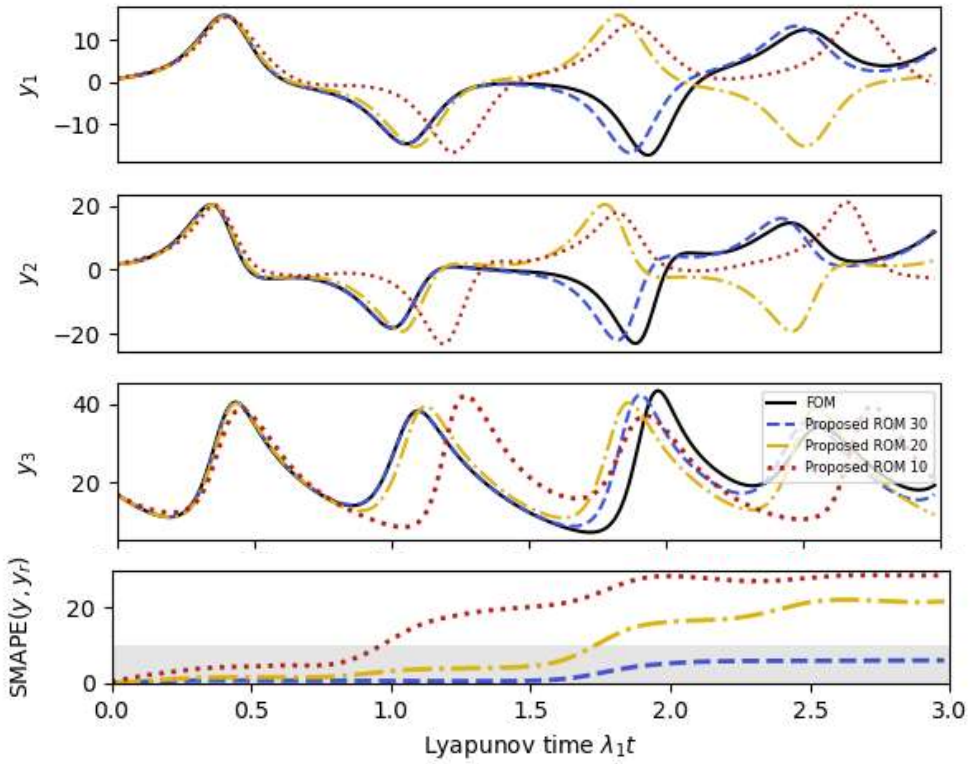
$$\dot{x}_r = \Psi^T f(\Phi x_r), \quad x_r(0) = \Psi^T x(0), \quad y_r = g(\Phi x_r)$$

for which FOM output $y(t)$ is approximated by ROM output $y_r(t)$ according to some performance metric (e.g. minimizing square error)

Skipping ahead: model reduction of a trained ESN

We can reduce the dimension of a trained ESN by as much as 85% with linear projections

In order to do this, we have to synthesize information about model sensitivity with information about variance of model trajectories, informed by simulation data



Stable LTI systems: balanced truncation

Model reduction for linear, time-invariant (LTI), minimal (controllable and observable) control systems is classically tackled using balanced truncation [1,2]

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Controllability Gramian: $W_c = \int_0^\infty e^{At} B B^T e^{A^T t} dt$, minimal control energy to drive state to the origin $x_0 W_c^{-1} x_0$

Observability Gramian: $W_o = \int_0^\infty e^{A^T t} C^T C e^{At} dt$, energy of the observable output with no control $x_0^T W_o x_0$

$$AW_c + W_c A^T = -BB^T \qquad A^T W_o + W_o A = -C^T C$$

[1] Moore B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*. 1981
[2] Glover K. All optimal Hankel-norm approximations of linear multivariable systems and their L,∞-error bounds. International journal of control. 1984

Stable LTI systems: balanced truncation

$$\dot{x} = Ax + Bu \qquad W_c = \int_0^\infty e^{At} B B^T e^{A^T t} dt \qquad W_o = \int_0^\infty e^{A^T t} C^T C e^{At} dt$$
$$y = Cx$$

The product $W_c W_o$ reflects controllability *and* observability; its eigenvalues are invariant under coordinate transformations, are singular values of Hankel operator.

Balanced truncation:

1. Find *balancing* coordinate transformation $\tilde{x} = Tx$ for which
$$W_c = W_o = \Sigma = \mathrm{diag}(\sigma_i)$$
2. *Truncate* states along the least controllable and least observable directions

[1] Moore B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*. 1981
[2] Glover K. All optimal Hankel-norm approximations of linear multivariable systems and their L,∞-error bounds. International journal of control. 1984

Unstable LTI systems: balanced truncation with generalized Gramians

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

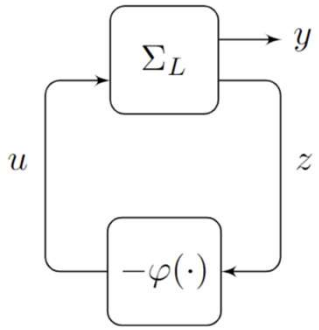
Analogous results can be established for unstable systems by defining the Gramians over the frequency domain [1]

$$W_c = \frac{1}{2\pi}\int_{-\infty}^{\infty}(i\omega I - A)^{-1}BB^T(-i\omega I - A^T)d\omega$$

$$W_o = \frac{1}{2\pi}\int_{-\infty}^{\infty}(-i\omega I - A^T)^{-1}C^TC(i\omega I - A)d\omega$$

[1] Zhou K, Salomon G, Wu E. Balanced realization and model reduction for unstable systems. International Journal of Robust and Nonlinear Control, 1999

Lur’e systems: balanced truncation?



A Lur’e system is a nonlinear system of the form

$$\dot{x} = Ax - B\varphi(C_z x)$$
$$y = C_y x$$

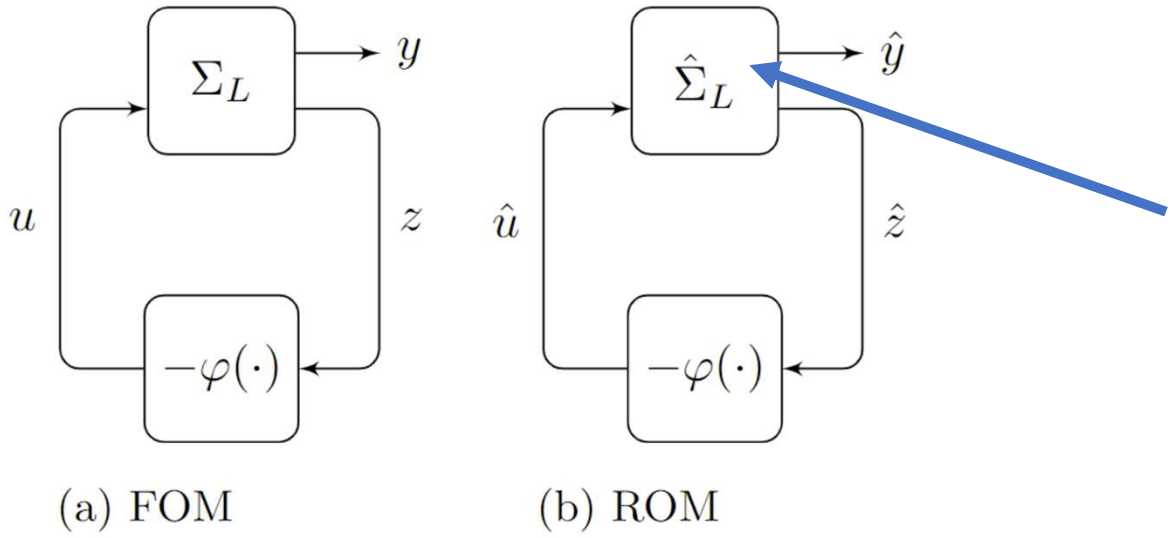
where the map φ satisfies some “nice” property (e.g. smoothness, boundedness, sector boundedness, contractivity). Lur’e systems can be considered a feedback interconnection of an LTI control system and a static map

$$\dot{x} = Ax + Bu \qquad\qquad u = -\varphi(z)$$
$$y = C_y x, \quad z = C_z x$$

Recent literature suggests that model reduction of the *nonlinear* Lur’e system is achievable by reducing its *linear* subsystem using balanced truncation [1]

[1] Padoan A, Forni F, Sepulchre R. Model reduction of dominant feedback systems. Automatica. 2021

Lur’e systems: balanced truncation?

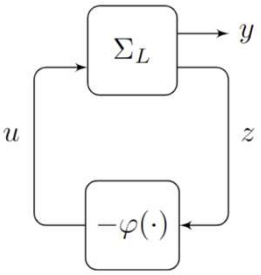


(a) FOM (b) ROM

General philosophy: all of the reduction is done here

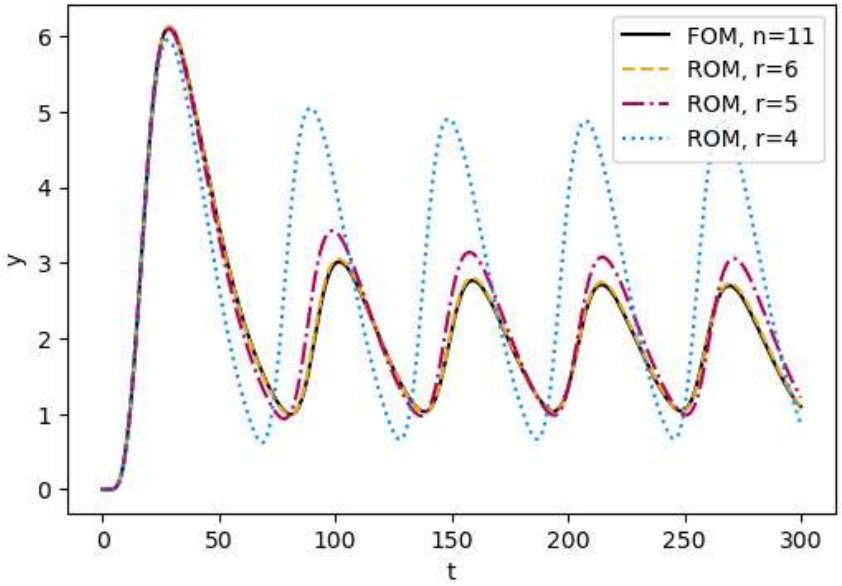
[1] Padoan A, Forni F, Sepulchre R. Model reduction of dominant feedback systems. Automatica. 2021

Example: network of Goodwin oscillators



$$\dot{x}_1 = \frac{1}{1+x_n^h} - \kappa_1 x_1$$
$$\dot{x}_i = x_{i-1} - \kappa_i x_i, \quad i = 2, \dots, n$$
$$y = x_n$$

Balanced truncation ROMs of Goodwin model



In Lur'e form, the model components are

$$A = \begin{bmatrix} -\kappa_1 & 0 & & \\ 1 & -\kappa_2 & 0 & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & 0 \\ & & & 1 & -\kappa_n \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

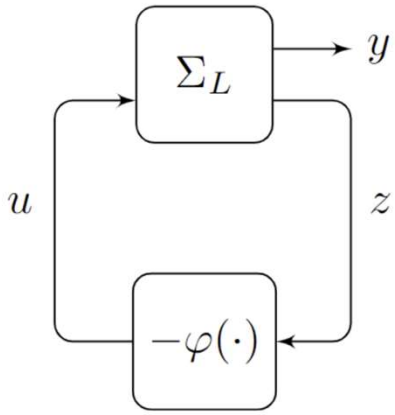
$$C = \begin{bmatrix} C_y \\ C_w \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 1 \end{bmatrix}^{\mathsf{T}}, \quad \varphi(w) = -\frac{1}{1+w^h} - \frac{\delta}{2} w$$

[1] Padoan A, Forni F, Sepulchre R. Model reduction of dominant feedback systems. Automatica. 2021

Trained echo state network as a Lur'e system

Consider a continuous-time ESN trained to forecast a low-dimensional dynamical system

$$\tau \dot{r} = -\lambda r + \tanh((W_r + W_{in}W_{out})r + b) \qquad y(t) = W_{out} r(t)$$



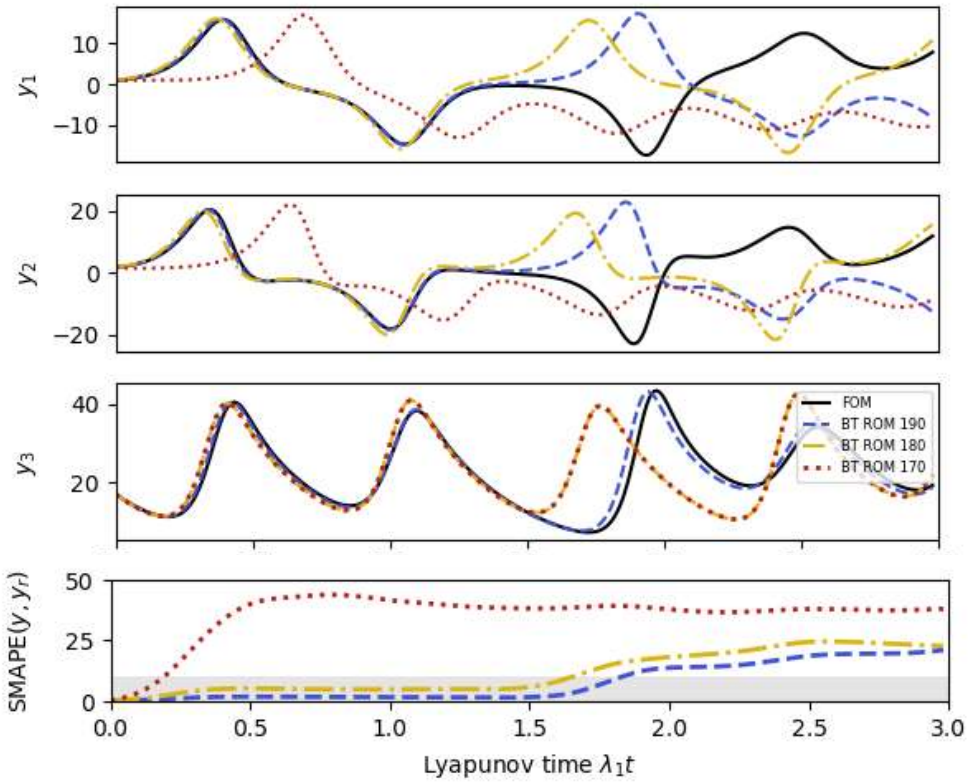
In Lur'e form, the system components are

$$A = \frac{1}{\tau}\left(-\lambda I + \frac{1}{2}(W_r + W_{in}W_{out})\right) \qquad B = \frac{1}{\tau} I$$

$$C_y = W_{out} \qquad C_z = W_r + W_{in}W_{out}$$

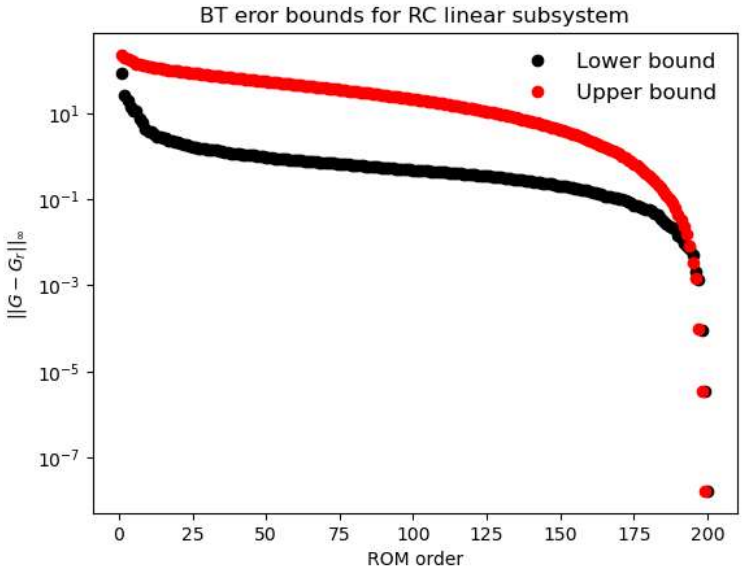
$$\varphi(z) = -\tanh(z + b) + \frac{1}{2} z$$

Trained ESN: balanced truncation is ineffective



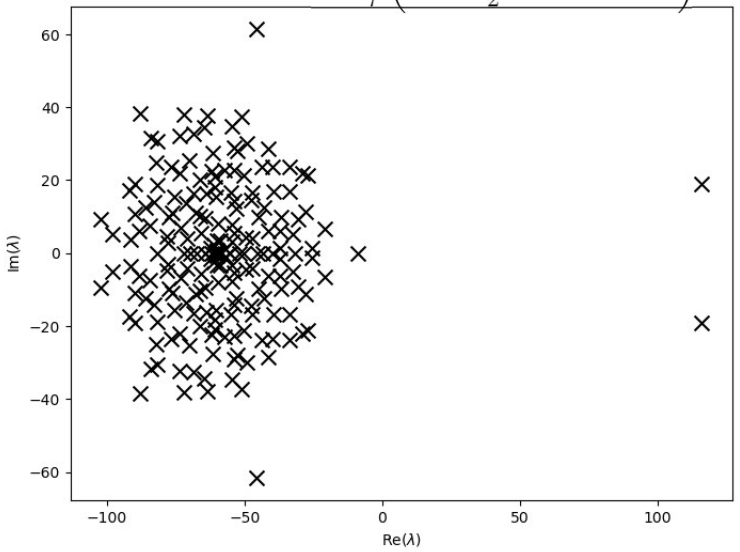
Reducing state dimension by as little as 15% (truncating 30 of the 200 states) results in a ROM that diverges from FOM almost immediately!

Balanced truncation failing isn't surprising



For LTI system that can be reduced significantly, error bounds should be less flat

Eigenvalues of $A = \frac{1}{\tau}\left(-\lambda I + \frac{1}{2}(W_r + W_{in}W_{out})\right)$



Spectrum of A suggests there are few dominant modes in linear subsystem, but many energetic modes as a consequence of random initialization during training

Data-informed balancing

Balancing is a good philosophy – both sensitivity and controllability matter!
Need an alternative to the Gramian matrices that captures local information

Consider the nonlinear FOM $\dot{x} = f(x), \quad x(0) = x_0, \quad y = g(x), \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m$

And the map $F : \mathbb{R}^n \to \mathcal{F}[0, T]$

$$x_0 \mapsto y(t), t \in [0, T]$$

Suppose $P \in \mathbb{R}^{n \times n}$ is a rank-r projection yielding the decomposition

$$x = x_1 + x_2, \quad x_1 = Px, \quad x_2 \in Null(P)$$

[1] Otto SE, Padovan A, Rowley CW. Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. *SIAM Journal on Scientific Computing*. 2023

Data-informed balancing

$$F : \mathbb{R}^n \to \mathcal{F}[0, T]$$
$$x_0 \mapsto y(t), t \in [0, T]$$

The optimal estimate (in the mean square sense) for $F(x)$ given Px is found by averaging over x_2:

$$\hat{F}(Px) = \mathbb{E}\left[F(x_1 + x_2) \;\middle|\; x_1\right](Px).$$

x_2 should have small variance and F should not be sensitive to variations in x_2.

[1] Otto SE, Padovan A, Rowley CW. Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. *SIAM Journal on Scientific Computing*. 2023

Data-informed balancing

Variance about the origin is quantified using the state covariance

$$W_x = \mathbb{E}\left[xx^T\right].$$

Sensitivity of F is quantified using the gradient covariance

$$W_g = \mathbb{E}\left[\nabla F(x)\nabla F(x)^T\right], \qquad \nabla F(x) = \mathrm{D}\,F(x)^T.$$

[1] Otto SE, Padovan A, Rowley CW. Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. *SIAM Journal on Scientific Computing*. 2023

Lur’e system state covariance

$$\dot{x} = Ax + Bu$$
$$y = C_y x, \quad z = C_z x$$
$$u = -\varphi(z)$$

Consider a bounded smooth trajectory $x(t), \quad x(0) = x_0, \quad t \in [0, T]$

Define $W_x = \frac{1}{T} \int_0^T x(t)x(t)^T dt$

Over the frequency domain this becomes

$$W_x = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(\omega_1 - \omega_2)T/2} \operatorname{sinc}\left((\omega_1 - \omega_2)T/2\right) (i\omega_1 I - A)^{-1} B \tilde{U}(i\omega_1) \tilde{U}^T(-i\omega_2) B^T (-i\omega_2 I - A^T)^{-1} d\omega_1 d\omega_2$$

Lur’e system state covariance

$$\dot{x} = Ax + Bu$$
$$y = C_y x, \quad z = C_z x$$
$$u = -\varphi(z)$$

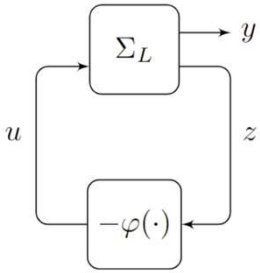
This is related to the generalized Gramian! In the infinite time limit,

$$\lim_{T\to\infty} TW_x = \frac{1}{2\pi}\int_{-\infty}^{\infty} (i\omega I - A)^{-1} B\tilde{U}(i\omega)\tilde{U}^T(-i\omega)B^T(-i\omega I - A^T)^{-1} d\omega$$

Recall the controllability Gramian was defined as

$$W_c = \frac{1}{2\pi}\int_{-\infty}^{\infty} (i\omega I - A)^{-1} BB^T(-i\omega I - A^T) d\omega$$

Lur’e system gradient covariance



In [1] gradient covariance is computed from data using ensembles of adjoint simulations. This is too much work for a Lur’e system! In the limit of infinite time, empirical gradient covariance for Lur’e system converges to generalized observability Gramian

$$W_o = \frac{1}{2\pi}\int_{-\infty}^{\infty}(-i\omega I - A^T)(C_y^T C_y + C_z^T C_z)(i\omega I - A)d\omega$$

$$\dot{x} = Ax - B\varphi(C_z x)$$
$$y = C_y x$$

vs.

$$\dot{x} = Ax + Bu$$
$$y = C_y x, \quad z = C_z x$$
$$u = -\varphi(z)$$

[1] Otto SE, Padovan A, Rowley CW. Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. *SIAM Journal on Scientific Computing*. 2023

Balancing nonlinear state covariance and linear observability

$$\dot{x} = Ax + Bu \qquad u = -\varphi(z) \qquad W_x = \frac{1}{T}\int_0^T x(t)x(t)^T dt$$
$$y = C_y x, \quad z = C_z x$$

The product $W_x W_o$ reflects (local) controllability *and* observability; its eigenvalues are invariant under coordinate transformations.

Balancing model reduction:

1. Estimate W_x using simulated data snapshots, W_o from Lyapunov equations
2. Find *balancing* coordinate transformation $\tilde{x} = Tx$ for which

$$W_x = W_o = \Sigma = \mathrm{diag}(\sigma_i)$$

3. *Truncate* states along the least controllable and least observable directions

[1] Moore B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*. 1981
[2] Glover K. All optimal Hankel-norm approximations of linear multivariable systems and their L,∞-error bounds. International journal of control. 1984

Practical computation procedure

Collect state snapshot data $X = [x(0), x(1), ..., x(N)]$

Factor $W_o = YY^T$ (e.g. Cholesky decomposition)

Compute the SVD: $Y^T X = U\Sigma V^T$.

Construct $\Phi = XV_r\Sigma_r^{-1/2}$ and $\Psi = YU_r\Sigma_r^{-1/2}$

This is equivalent to doing POD in a weighted inner product space [1]!

Inherited error bound:

$$\int_0^T \|x(t) - x_r(t)\|_{W_o}^2 \leq k \sum_{i=r+1}^{n} \left(\sigma_i + \int_0^T |x(t)^T A^T + \varphi(C_y x(t))^T B^T W_o \psi_i| dt \right)$$

[1] Volkwein S. Proper orthogonal decomposition: Theory and reduced-order modelling. Lecture Notes, University of Konstanz. 2013

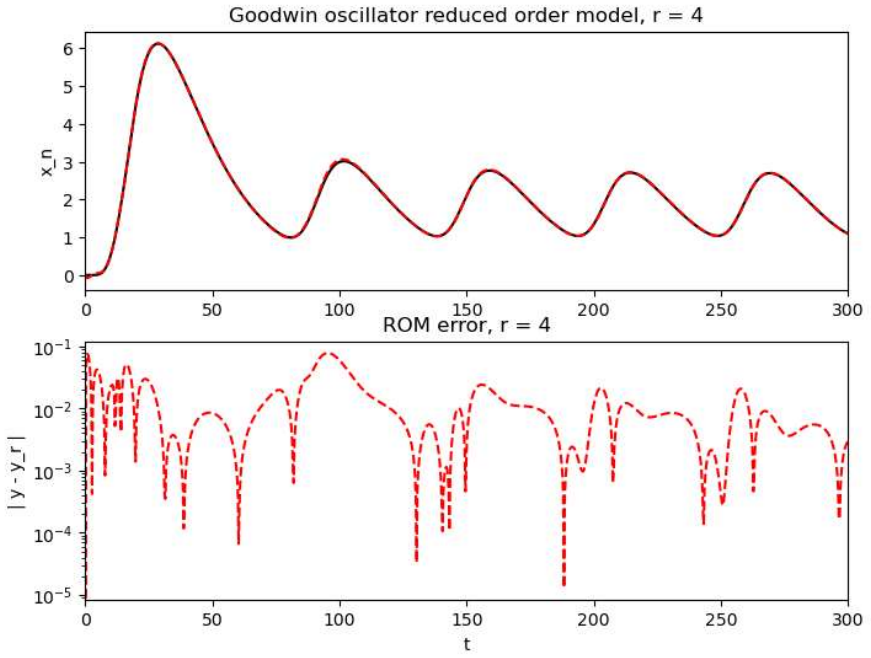
Network of Goodwin oscillators

Can reduce the 11-node Goodwin model further than the balanced truncation-based approach

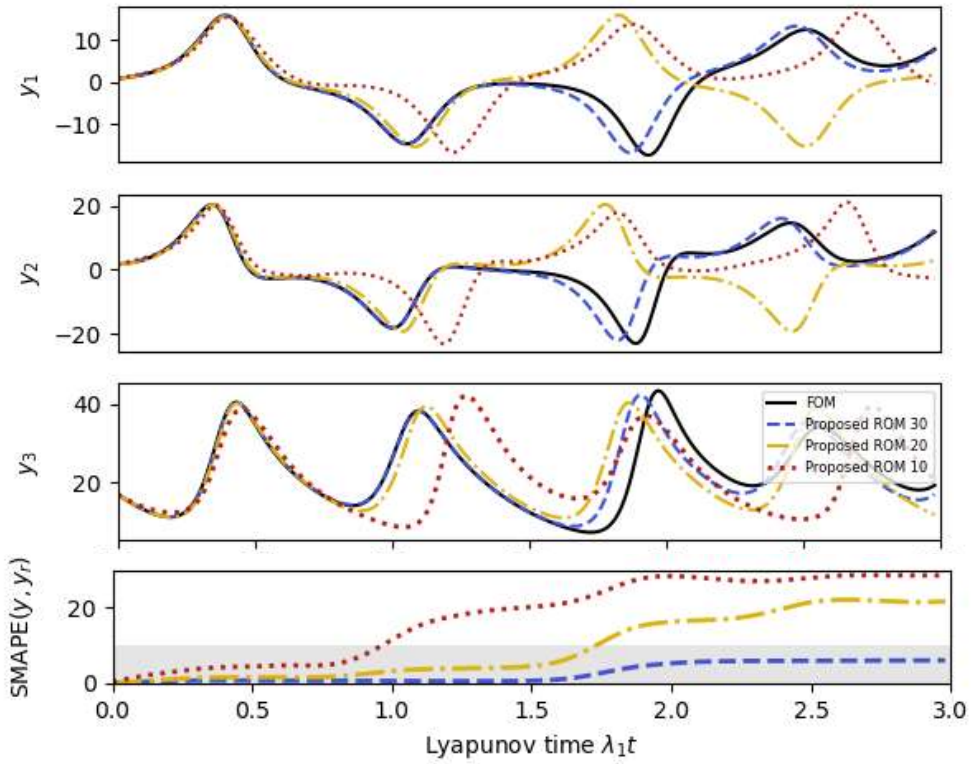
$$A = \begin{bmatrix} -\kappa_1 & 0 & & \\ 1 & -\kappa_2 & 0 & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & 0 \\ & & & & 1 & -\kappa_n \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} C_y \\ C_w \end{bmatrix} = \begin{bmatrix} 0 \cdots 0\ 1 \\ 0 \cdots 0\ 1 \end{bmatrix}^{\mathsf{T}},$$

$$\varphi(w) = -\frac{1}{1+w^h} - \frac{\delta}{2}w$$



Trained ESN



Reduction of state space by as much as 85% recovers a good ROM, while as many as 95% of the total states (190 of the 200) can be truncated with the ROM still producing an accurate short-term forecast and accurate attractor statistics

Summary

- We extended a data-informed model reduction approach of Otto et. al. for feedback systems in Lur'e form
  - Expensive gradient adjoint computations are not necessary in this setting
  - Sensitivity is captured by standard observability Gramian, can be computed efficiently from a set of algebraic Lyapunov equations
  - Method is equivalent to finding a good inner product space for POD
- We gained some insights about echo state networks
  - Low-dimensional nonlinear model is indeed recoverable from the trained dynamics
  - Observability Gramian induces a natural inner product space for the solutions of trained ESNs; useful to think about these objects from this perspective