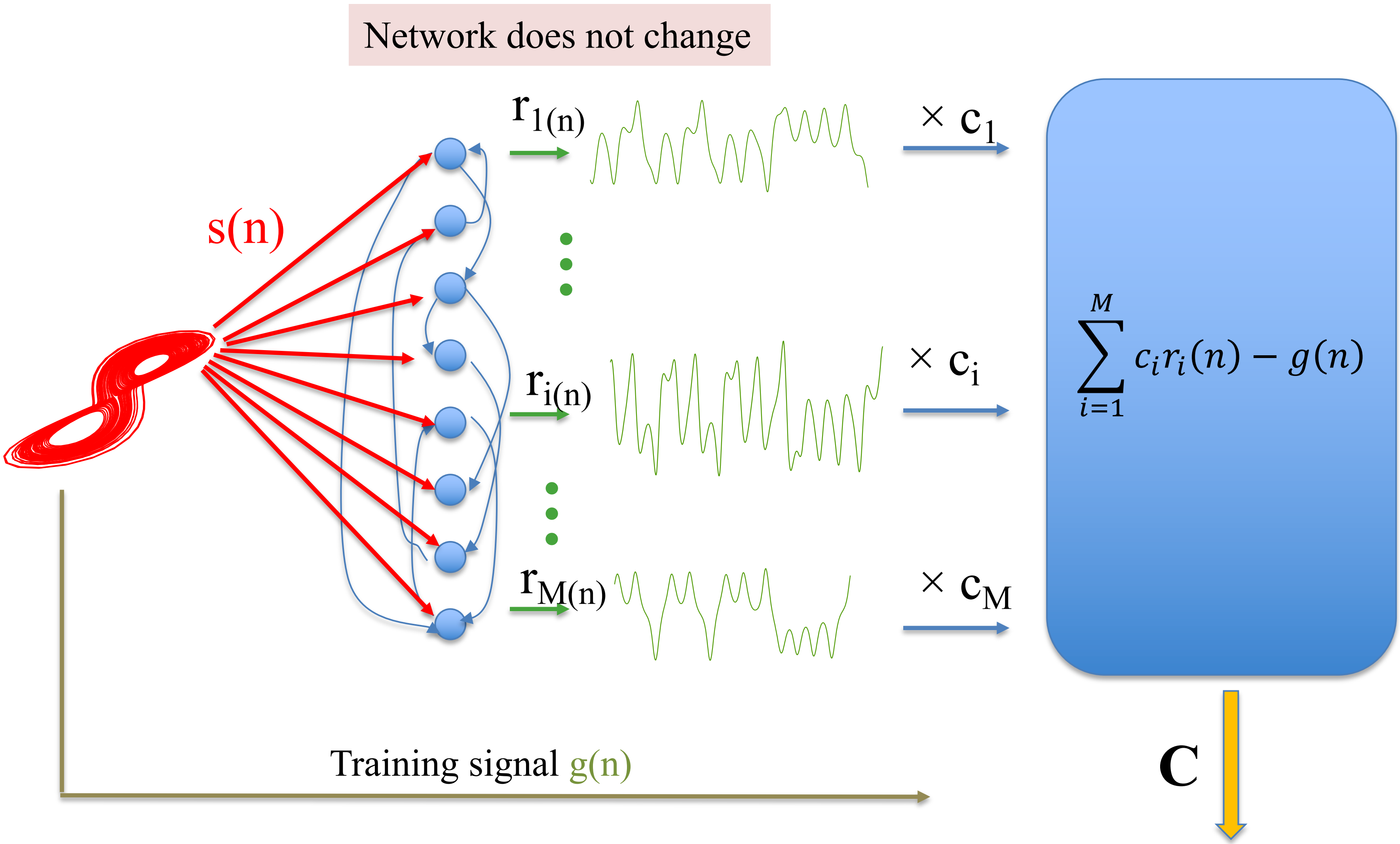
Time Shifts to Reduce the Size of Reservoir Computers

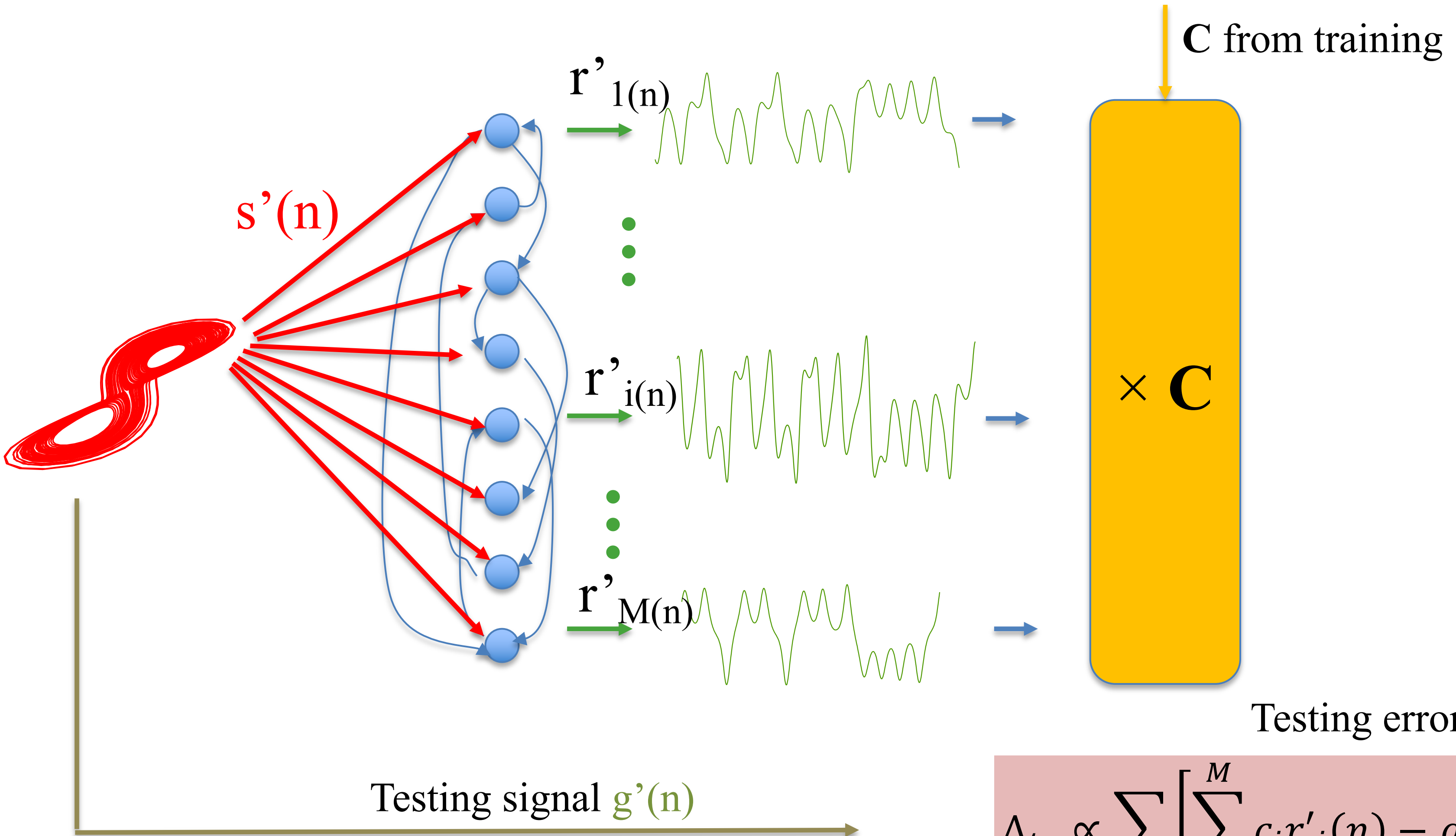
Tom Carroll, Code 6392, thomas.carroll@nrl.navy.mil
Joe Hart, Code 5675

US Naval Research Lab
Washington, DC 20375
USA

Training


Network does not change
s(n)
r1(n)
ri(n)
rM(n)
× c1
× ci
× cM
$\sum_{i=1}^{M} c_i r_i(n) - g(n)$
Training signal g(n)
C

Testing



$$\Delta_{tx} \propto \sum_{n} \left[\sum_{i=1}^{M} c_i r'_i(n) - g'(n) \right]$$

U.S. NAVAL RESEARCH LABORATORY

$$\mathbf{R}(n+1) = f\left[\mathbf{A}\mathbf{R}(n) + \mathbf{W}s(n)\right]$$

R is vector of reservoir variables
A is adjacency matrix: how are nodes connected
W is vector of input coefficients
s is input signal

- Training involves far fewer parameters than neural network.
- Simple linear fit: no stability concerns.

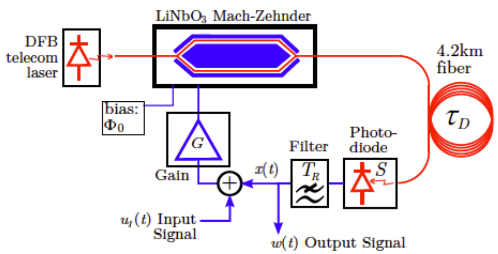
With no need to train internal connections: many constraints in designing neural networks no longer apply.

- Node types can vary (not just sigmoid)
- Can be built from analog hardware

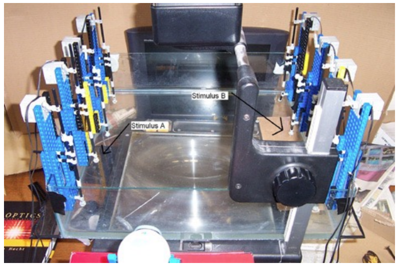
Reservoir computers are built from analog nodes

About 100 to 1000 nodes

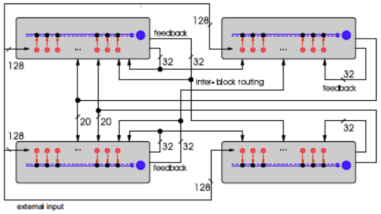
optical



Fluid system



Analog electronics



- Neural tissue
- Field programmable gate arrays
- E. coli gene regulation network

Reservoir computer requirements

Operate near edge of stability

Not always

Maximize memory

Must tune memory for particular task; can have too much

Sigmoid activation function

Other nonlinearities also work

Sparse connection matrix

Yes but must maintain strong interaction between nodes

High dimensional

What is it about high dimensional space that makes a reservoir computer work?

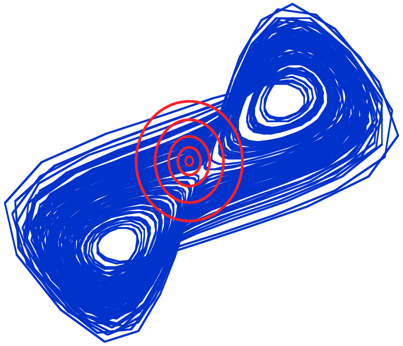
Consideration:

Reservoir computers are not general purpose computers. They are universal function approximators

L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Neural Networks,* vol. 108, pp. 495-508, 2018/12/01/ 2018.

Dimension

Correlation dimension $n(r) = r^d$



Difficult with data
Low dimensional only

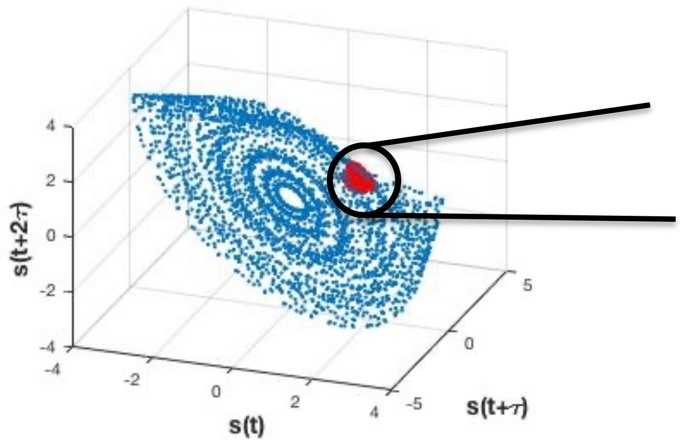
False nearest neighbor (FNN) dimension

1) Embed data in d dimensions
2) Find neighbors
3) Embed in d+1 dimensions
4) How many points are still neighbors?
• If most points still neighbors, increase d

<u>FNN requires an arbitrary threshold</u>

Covariance Dimension

Embedded signal



$\mathbf{s}_c(t)$ n points d dimensions

Subtract mean, set to unit norm: $\mathbf{s}_{cn}(t)$

$$\mathbf{C} = \frac{\mathbf{s}_{cn}^T \mathbf{s}_{cn}}{n}$$

Covariance matrix

d × d matrix

<u>Compare to covariance for random process.</u>

- Covariance matrix eigenvalues indicate probability that embedded signal is anisotropic in d dimensions- assume this means it can be embedded

T. L. Carroll and J. M. Byers, "Dimension from covariance matrices," *Chaos,* vol. 27, p. 023101, Feb 2017.

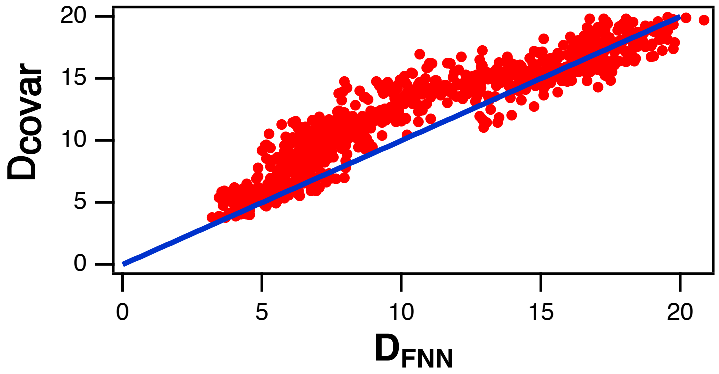
Dimension

Covariance and FNN dimensions can be adapted for reservoir computers

T. L. Carroll, "Dimension of reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 30, p. 013102, 2020.

T. L. Carroll, "Low dimensional manifolds in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 31, p. 043113, 2021.

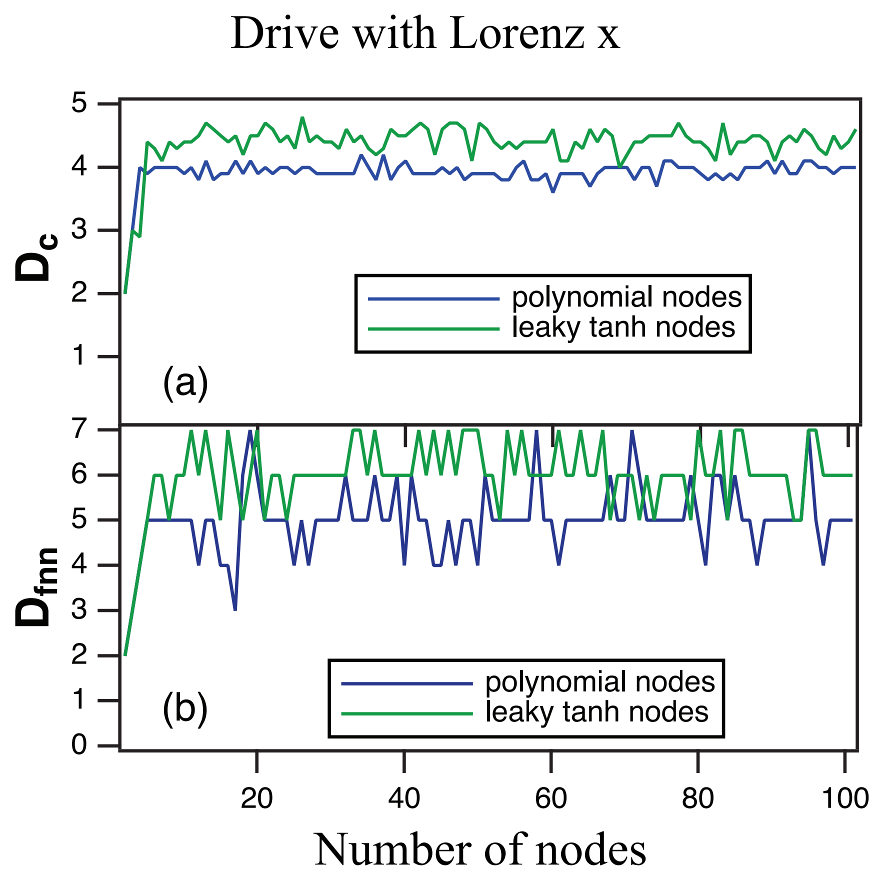
Covariance dimension measures geometry: FNN dimension measures predictability



Dimension measurements for reservoirs with random parameters

Both methods give similar dimensions

Reservoir computer dimensions



Reservoir signals live on a low dimensional manifold

Measuring signal diversity: covariance rank

Similar to T. Lymburn et al., *Chaos,* vol. 29, p. 023118, 2019/02/01 2019.

Put reservoir time series signals into matrix Ω

$$\Omega = \begin{bmatrix} r_1(1) & \cdots & r_M(1) \\ r_1(2) & & r_M(2) \\ \vdots & & \vdots \\ r_1(N) & \cdots & r_M(N) \end{bmatrix}$$

How many orthogonal directions are there in Ω?

- Use PCA: subtract mean, normalize, find covariance matrix Θ

$$\Theta = \Omega_{norm}^T \Omega_{norm}$$

Eigenvectors of Θ are orthogonal directions: find rank of Θ

$$\Gamma = \mathrm{rank}(\Theta)$$

Dimension vs. Rank

High rank does not mean high dimension; same for the other way

Create a matrix of signals from Lorenz x signal with different initial conditions

$$\begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_{100}(t) \\ x_1(t+\tau) & x_2(t+\tau) & & x_{100}(t+\tau) \\ \vdots & \vdots & & \vdots \\ x_1(t+(N-1)\tau) & x_2(t+(N-1)\tau) & \cdots & x_{100}(t+(N-1)\tau) \end{bmatrix}$$

Rank of covariance matrix = 100
Covariance dimension = 5
False nearest neighbor dimension = 8

Signals and Reservoirs

Lorenz

$$\frac{dx}{dt} = T_l\left(p_1\left(y - x\right)\right)$$
$$\frac{dy}{dt} = T_l\left(x\left(p_2 - z\right) - y\right)$$
$$\frac{dy}{dt} = T_l\left(xy - p_3 z\right)$$

Rossler

$$\frac{dx}{dt} = T_r\left(-y - p_1 z\right)$$
$$\frac{dy}{dt} = T_r\left(x + p_2 y\right)$$
$$\frac{dz}{dt} = T_r\left(p_3 + z\left(x - p_4\right)\right)$$

Drive with x signals, train on z signals

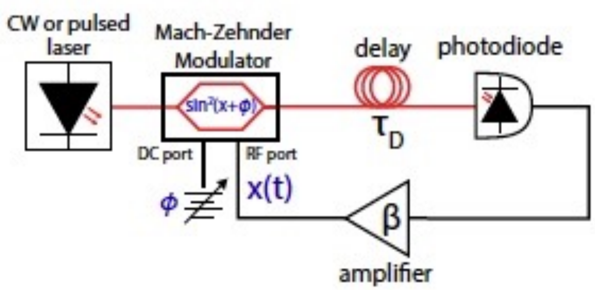
Tanh map reservoir computer

$$\mathbf{R}\left(n+1\right) = g\tanh\left(\mathbf{AR}\left(n\right) + \varepsilon s\left(n\right) + 1\right)$$

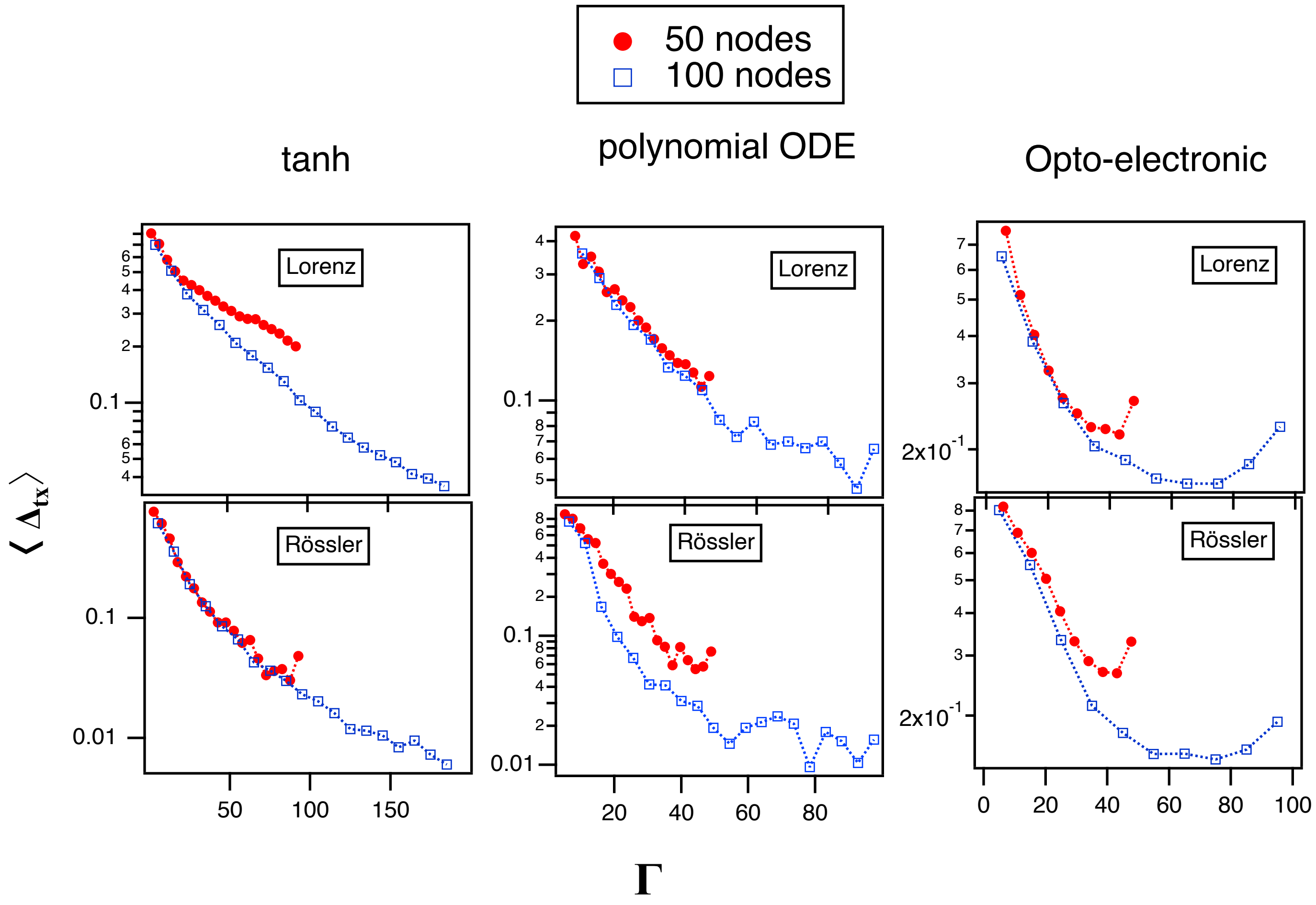
Polynomial ODE reservoir computer

$$\frac{dr_i(t)}{dt} = \alpha\left[p_1 r_i\left(t\right) + p_2 r_i^2\left(t\right) + p_3 r_i^3\left(t\right) + \sum_{j=1}^{M} A_{ij} r_j\left(t\right) + W_i s\left(t\right)\right]$$

Laser delay reservoir computer



Many different reservoirs: error vs rank



Distribution Statement A. Approved for public release. Distribution unlimited.

Memory capacity (Jaeger)

Does reservoir computer remember a noise signal?

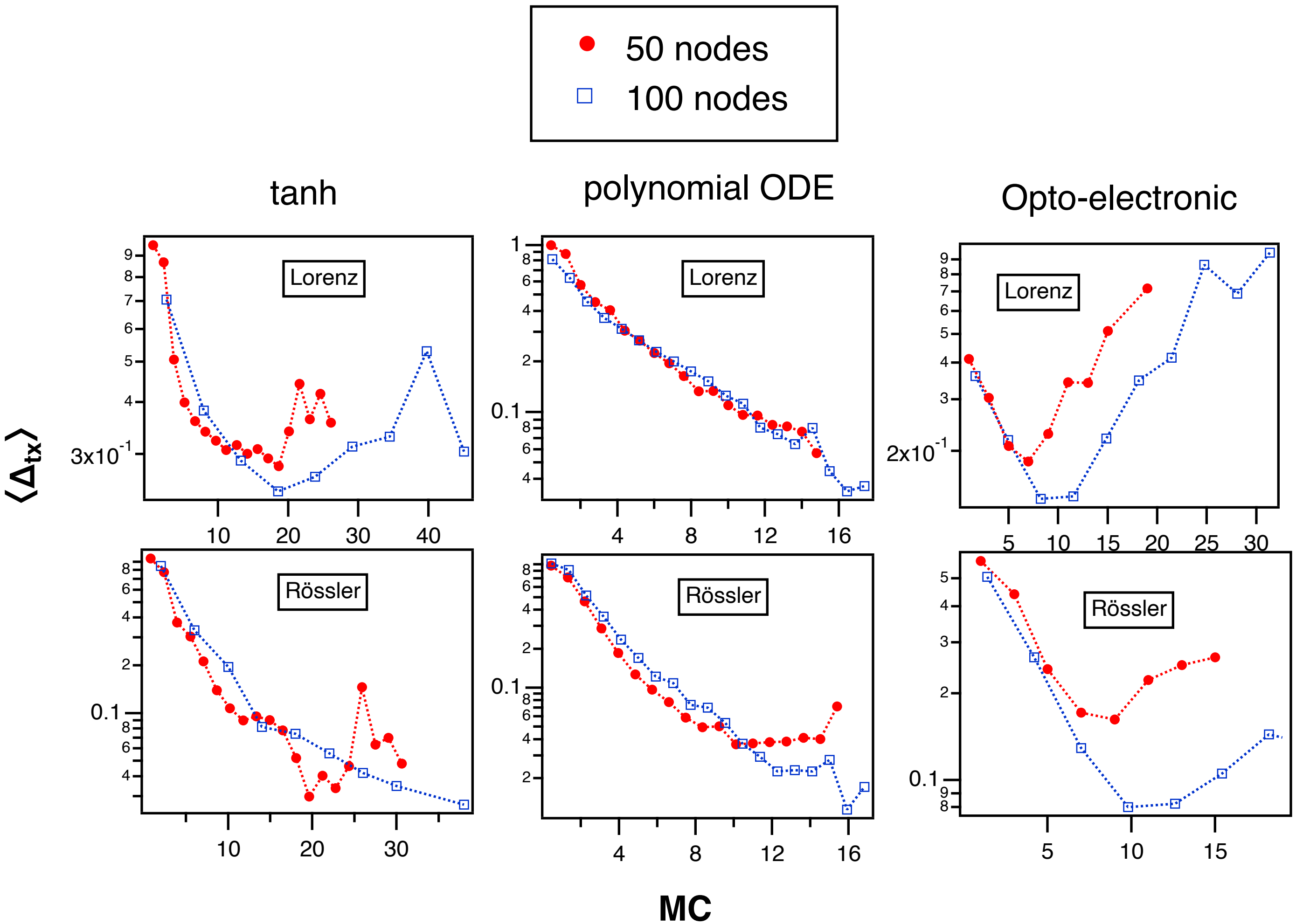
- Drive with noise signal *s(n)*
- train on *s(n-τ)*
- fit is $g(n)=\sum_i c_i r_i(n)$

Memory capacity is cross correlation between *s(n-τ)* and *g(n)*

$$\mathrm{MC}_k = \frac{\sum_{n=1}^{N} [s(n-k)-\overline{s}][g_k(n)-\overline{g_k}]}{\sum_{n=1}^{N} [s(n-k)-\overline{s}] \sum_{n=1}^{N} [g_k(n)-\overline{g_k}]}$$

Warning: in use, reservoir is not driven with noise. Results may be incorrect

Many different reservoirs: error vs memory



Reservoir computer performance depends on rank or memory

- ***Independent of number of nodes***

How to increase rank or memory

- *Do we really need many nodes?*

Conventional wisdom is that reservoir computers work because of high dimension

Really, they work because of high rank or memory- is high dimension necessary?

Use ideas from E. D. Frate, A. Shirin, and F. Sorrentino, "Reservoir computing with random and optimized time-shifts," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 31, p. 121103, 2021.

Increasing Rank and Memory

Adapt this for our use:

- Ordered set of time shifts
- Reservoir has M_1 nodes: create $M_2 \geq M_1$ time shifts

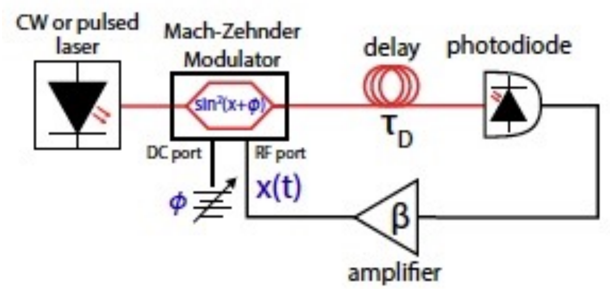
M_2 delays, but reservoir has only M_1 signals: example for M_1=3, M_2=9

$$\Omega_2 = \begin{bmatrix} r_1\left(1-\frac{\tau_{max}}{9}\right) & r_2\left(1-\frac{2\tau_{max}}{9}\right) & r_3\left(1-\frac{3\tau_{max}}{9}\right) & r_1\left(1-\frac{4\tau_{max}}{9}\right) & \cdots & r_3\left(1-\frac{9\tau_{max}}{9}\right) \\ r_1\left(2-\frac{\tau_{max}}{9}\right) & & & & & r_3\left(2-\frac{9\tau_{max}}{9}\right) \\ \vdots & & & & & \vdots \\ r_1\left(N-\frac{\tau_{max}}{9}\right) & \cdots & & & \cdots & r_3\left(N-\frac{9\tau_{max}}{9}\right) \end{bmatrix}$$

Use interpolation for non-integer time shifts

Laser Delay Reservoir Computer

Simulation and experiment use laser delay system



J. D. Hart et al, Phil Trans R. Soc A. **377** 20180123

$$f\left(x\left(t\right),s\left(t\right)\right)=\frac{1}{T_L}\left(-x\left(t\right)+\beta\sin\left(x\left(t-\tau_D\right)+\rho W_i s_{in}\right)\right)^2$$

Create virtual nodes by time delay: time per node θ, delay loop time τ_D

- number of nodes $M_1=\tau_D/\theta$
- Nodes coupled by low pass time constant T_L: one way ring network

$$\Omega_1=\begin{bmatrix}\nu\left(\theta\right) & \nu\left(2\theta\right) & \dots & \nu\left(M_1\theta\right)\\ \nu\left(\theta+\tau_D\right) & & & \vdots\\ \vdots & & & \vdots\\ \nu\left(\theta+N\tau_D\right) & \dots & \dots & \nu\left(M_1\theta+N\tau_D\right)\end{bmatrix}$$

U.S. NAVAL RESEARCH LABORATORY

Adding Time Shifts: Simulations

Time shifted matrix

$$\Omega_2\left(i,j\right)=\nu\left(k\theta+\left(i-1\right)\tau_D-\tau_j\right)$$

Input signals

Lorenz

$$\frac{dx}{dt}=T_l\left(p_1\left(y-x\right)\right)$$
$$\frac{dy}{dt}=T_l\left(x\left(p_2-z\right)-y\right)$$
$$\frac{dy}{dt}=T_l\left(xy-p_3z\right)$$

Rössler

$$\frac{dx}{dt}=T_r\left(-y-p_1z\right)$$
$$\frac{dy}{dt}=T_r\left(x+p_2y\right)$$
$$\frac{dz}{dt}=T_r\left(p_3+z\left(x-p_4\right)\right)$$

Drive with x signals, train on z signals

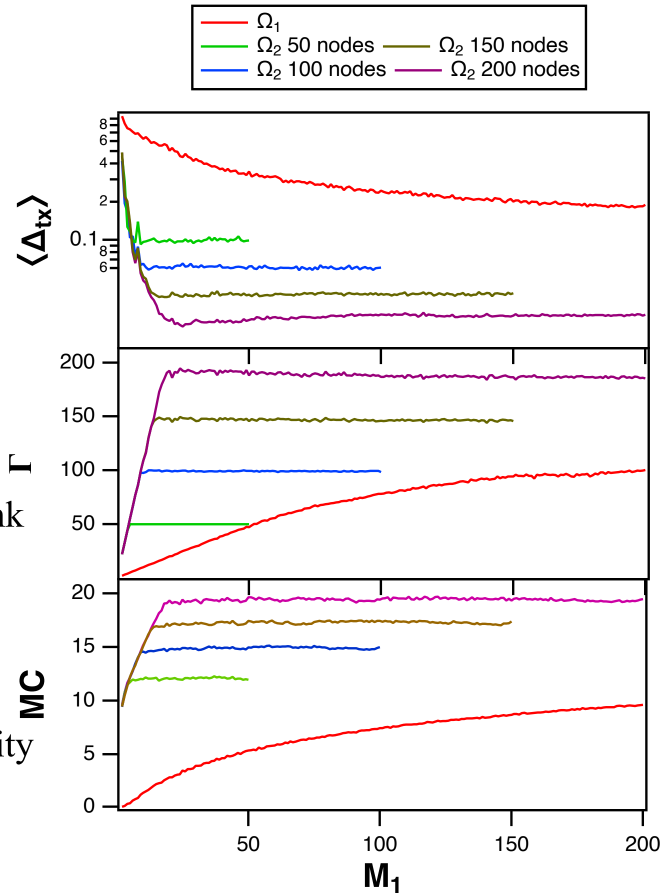
Laser Delay System: Simulations

Lorenz

Testing error

Covariance rank

Memory capacity



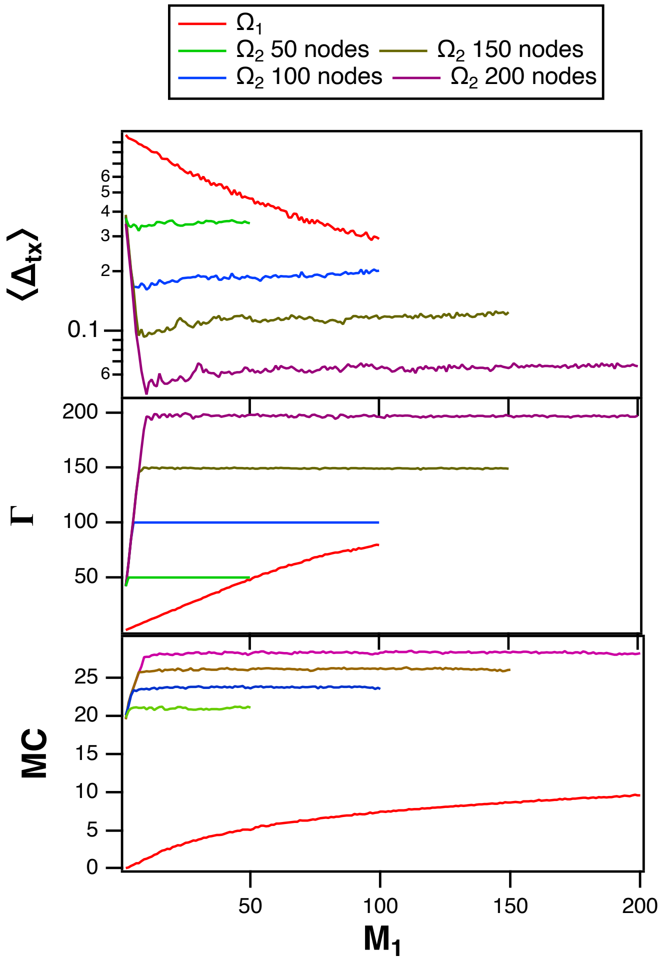
M_1 is size of original reservoir computer
M_2 is size of delay matrix

More delays:

- smaller testing error
- larger rank
- larger memory capacity

Adding a large delay matrix gives small testing error with small reservoir

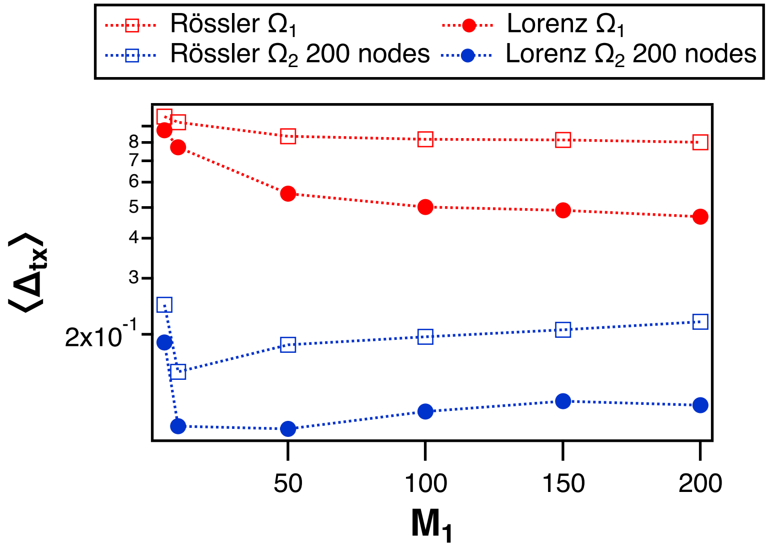
Laser Delay System: Simulations



Rössler: similar results to Lorenz

Same pattern with reservoir computers using tanh or polynomial nodes

Laser Delay System: Experiment



Adding time shifts decreases testing error

For laser delay system: more nodes means longer time delay:

- Adding time shifts increases speed

Known from work on nonlinear series approximations that nonlinear part and delay part can be separate

- Same is true here

Reservoir computer does not require high dimensions

- delays do not add dimension

This was suggested by measurements of low dimensional manifolds in reservoir computers

T. L. Carroll, "Dimension of reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 30, p. 013102, 2020.

T. L. Carroll, "Low dimensional manifolds in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 31, p. 043113, 2021.

Actual reservoir can be low dimensional: add rank and memory with delays

T. L. Carroll and J. D. Hart, "Time shifts to reduce the size of reservoir computers," *Chaos,* vol. 32, p. 083122