



Spiking Neural Networks for Control

Max Schaufelberger

February 13, 2024 — KTH Royal Institute of Technology



Table of Contents

Introduction

Simulation

Control

Learning

Combined Learning

Conclusion



Introduction

Goal / Motivation

Artificial SNN can already solve various cognitive task such as

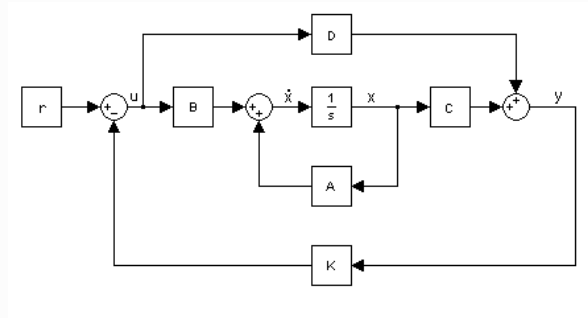
- Memorization
- Basic Logic
- Simulation of Dynamic Systems
- Control

Although with varying levels of biologic plausibility. We set out to build a controlled dynamic system based on SNN using learning and biologic plausibility

- Allow for black-box deployment without manual parameter tuning
- "Limit ourselves to use the brains capabilities to design a controller"

What are we talking about

Control a Linear system



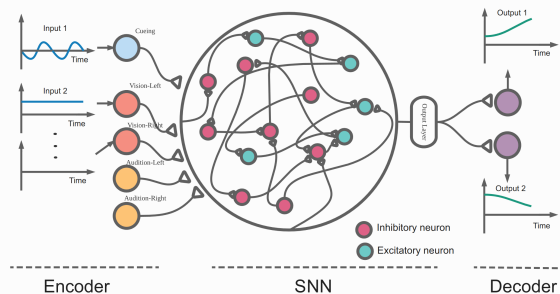
- Tracking of reference trajectory

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{1}$$

- Only stable systems

What are we talking about

Use Spiking neural networks



[Xue+22]

- Third Generation of NN
- Working with discrete spikes
- Inherently fit for temporal data

Method

1. Simulate

Use a spiking network to simulate a dynamic system

2. Control

Devise a control scheme to control the network output

3. Learn

Apply biologically plausible learning rules to our network

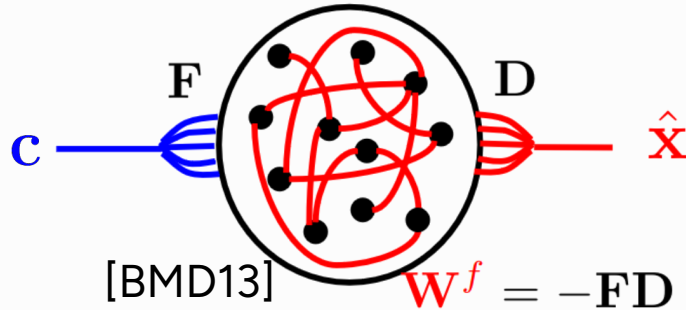
4. Combine

Integrate all three steps into a single controller



Simulation

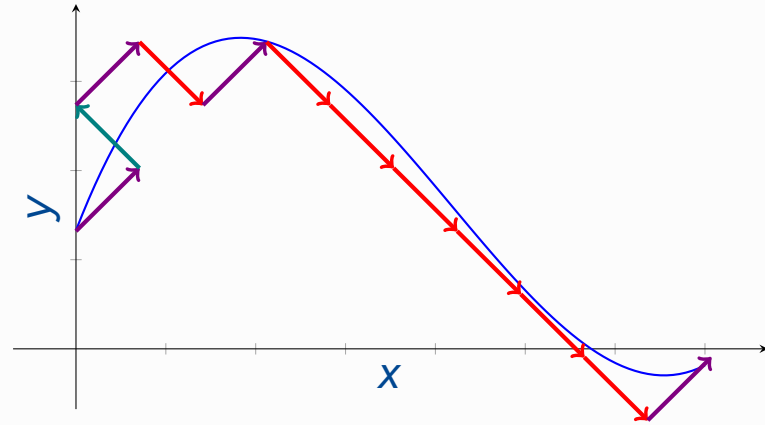
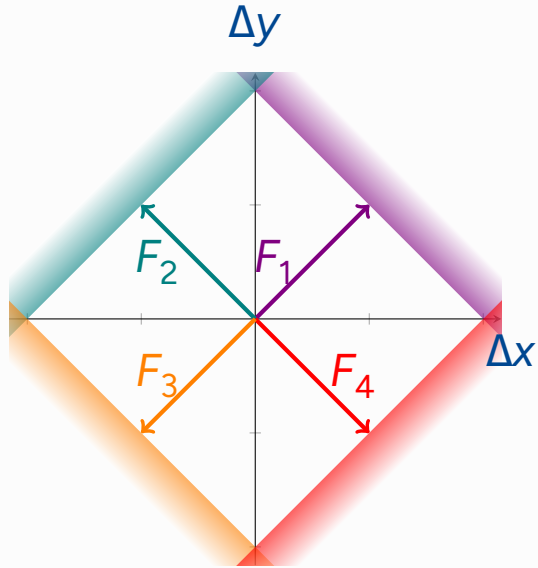
Simulation of Linear systems



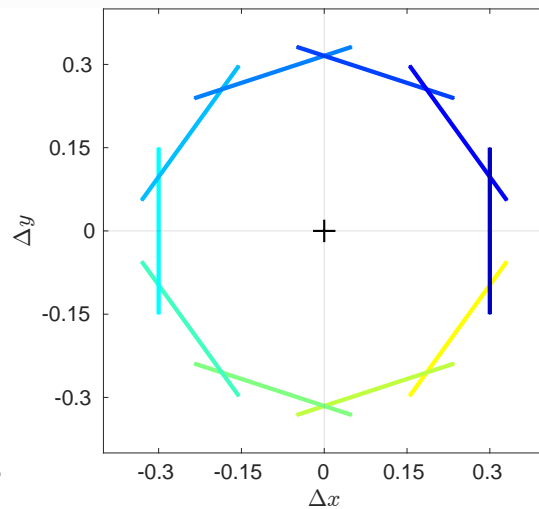
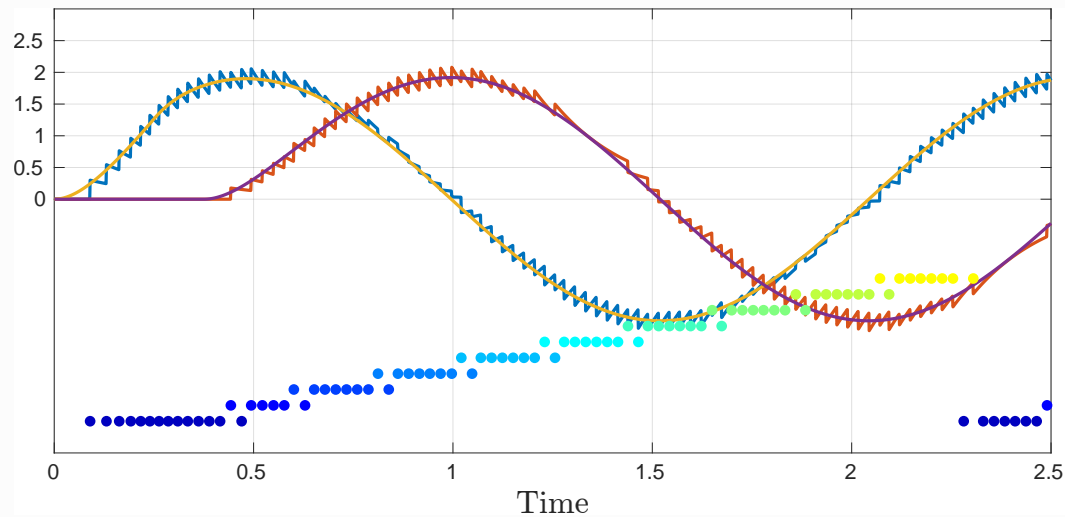
- Build NN that outputs \hat{x} from the system $\dot{x} = Ax + c$ given c
- Group of LIF neurons with with intrinsic Voltage, tracking the projected error $V_i = F_i(x - \hat{x}) + \mu r_i$
- Network decoding $\hat{x} = F^T r$

$$\dot{V} = -\lambda_V V + Fc + W^f o(t) + W^s r(t) + \sigma_V \eta(t)$$

Geometric



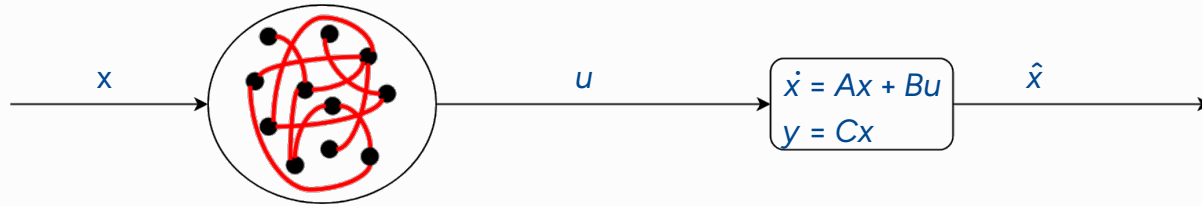
Example Simulation





Control

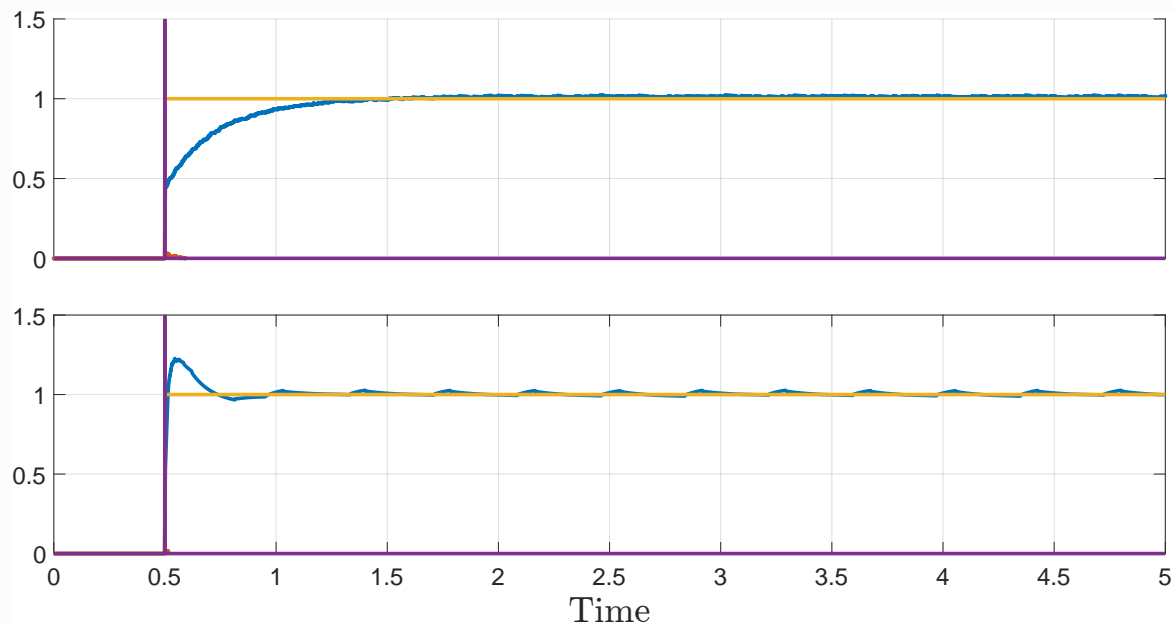
Control Concept



[HC19]

- (Almost) identical network architecture
- Network output is external input into (previous) simulating network \longleftrightarrow Network state contains control signal
- Governed by PD-control as $c = \dot{x} - Ax$
- In presence of output matrix $C \neq \mathbf{I} \leftrightarrow \text{rank}(B^T C^T) = \text{rank}(B^T)$

Example





Learning

Learning rules [BD15]

Slow Learning rule $W^s = F(A + \lambda_d \mathbf{I})F^T$

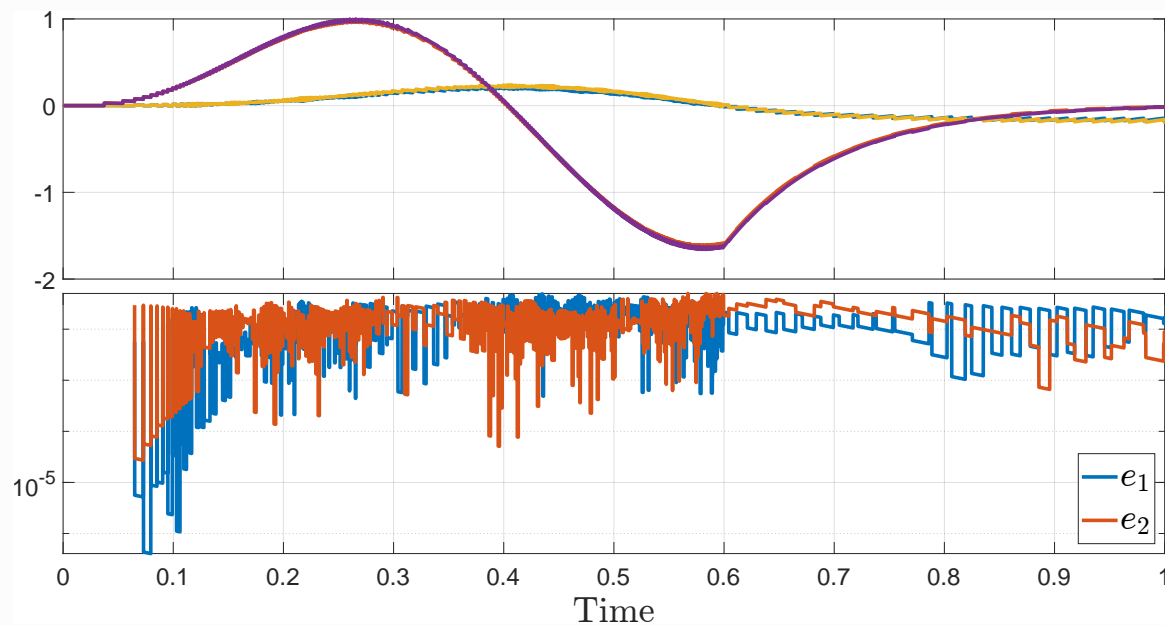
- Online Learning of Student teacher dynamics $\dot{\hat{x}} = M\hat{x} + c$
- Error Feedback Ke during Training
- $\delta M \propto e\hat{x}^T \longrightarrow \delta W^s \propto F(e\hat{x}^T)F^T \approx Fer^T$
- Supervised Learning rule

$$V_i = F_i(x - \hat{x}) - \mu r_i$$

Fast Learning rule $W^f = FF^T + \mu \mathbf{I}$

- Voltage measures system error
- Minimize average Voltage outside of Neuron Threshold
- Biologically plausible pre \times post locally
- Unsupervised Learning Rule

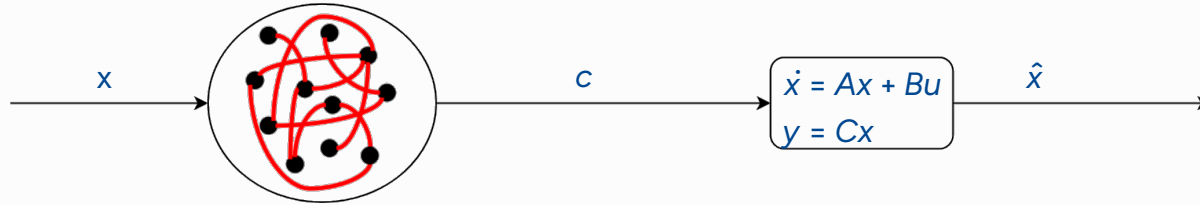
Example





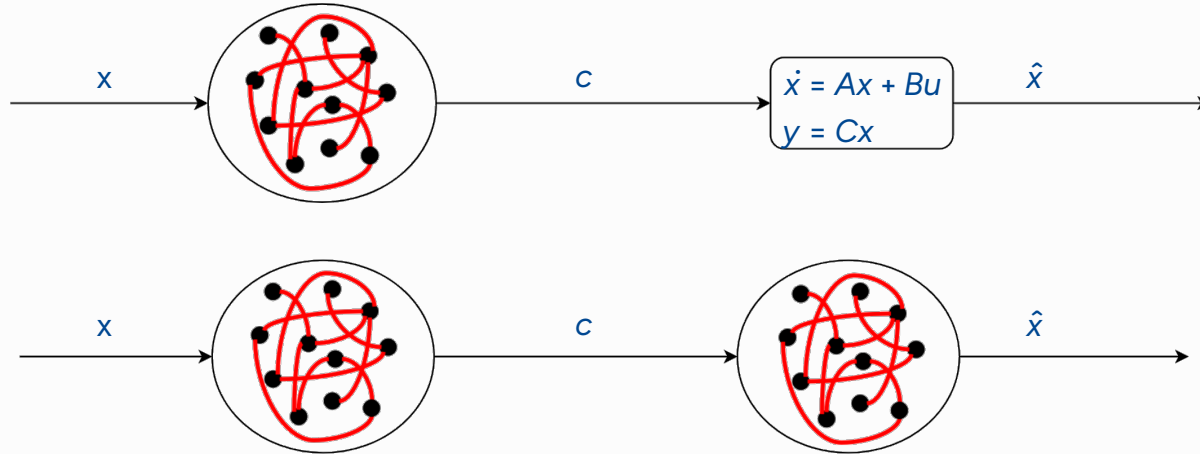
Combined Learning

Control Concept



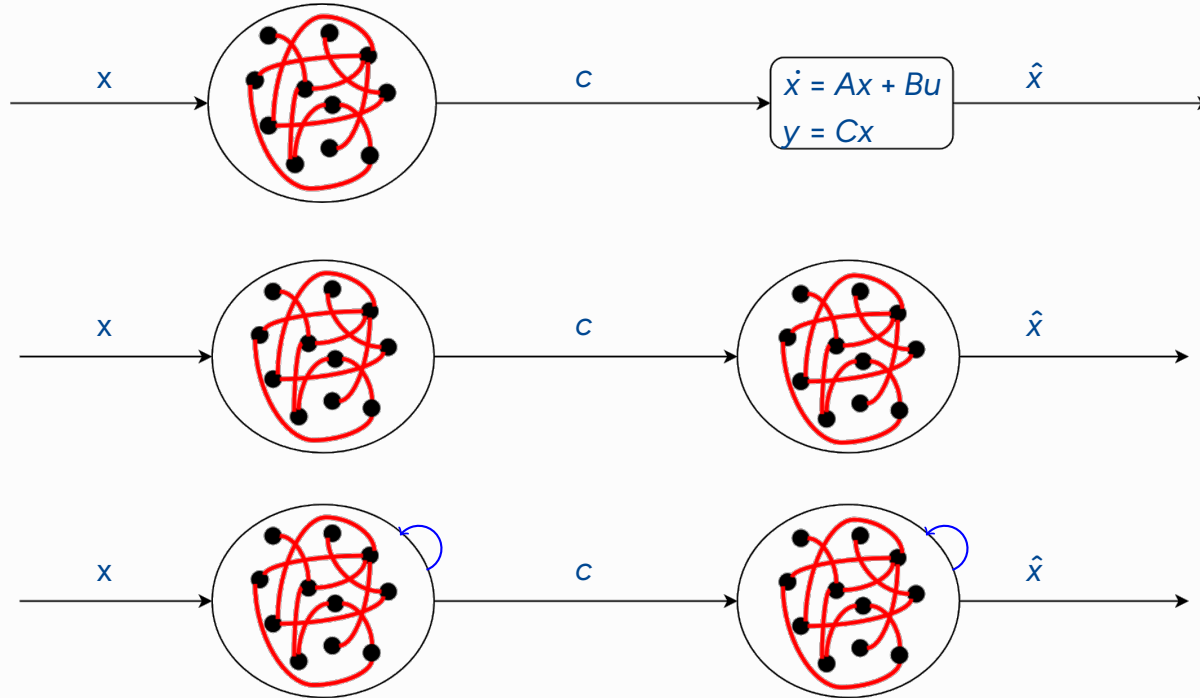
[HC19]

Control Concept



[HC19]

Control Concept



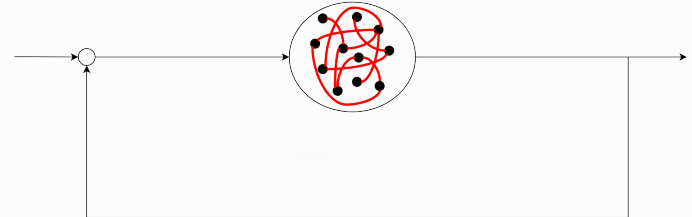
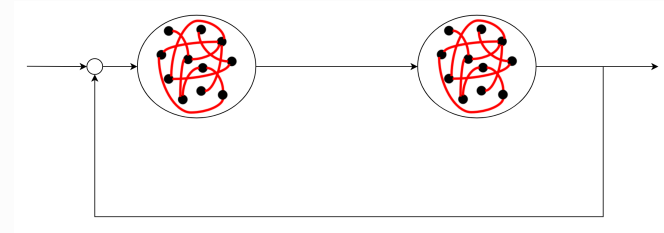
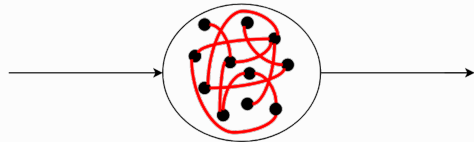
[HC19]

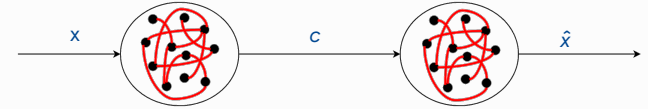
Problems

In conjunction, problems can arise:

- Divergence in Learning
- Control with Noise
- Open loop
- Biologically implausible Learning of first Network

Control Concept II





Dual Network

No Learning rule for control network available

Open Loop Control:

- Incapable of noise detection or correction
- No Compensation of Training error

Highly dependent on governing dynamics from $c_{\text{contr}} = \dot{x}_{\text{ref}} - Ax_{\text{ref}}$

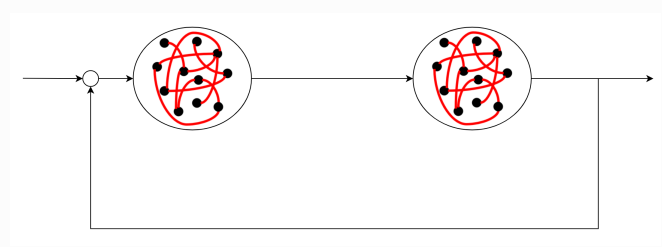
Dual Network with Feedback

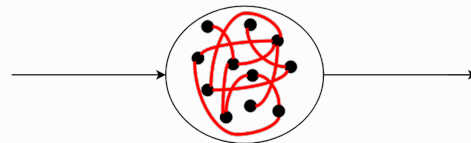
No Learning rule for control network available

~~Open loop Control:~~

- ~~• Incapable of noise detection or correction~~
- ~~• No Compensation of Training error~~

Highly dependent on governing dynamics from $C_{\text{contr}} = \dot{x}_{\text{ref}} - Ax_{\text{ref}}$





Single Network

~~No Learning rule for control network available~~

Open Loop Control:

- Incapable of noise detection or correction
- No Compensation of Training error

Highly dependent on governing dynamics from $\mathbf{c}_{\text{contr}} = \dot{\mathbf{x}}_{\text{ref}} - \mathbf{A}\mathbf{x}_{\text{ref}}$

Orthonormality restriction on Input Matrix $\mathbf{B} \in \mathbb{B} := \{\mathbf{M} \mid \mathbf{M}\mathbf{M}^T = \mathbf{I}\}$

Single Network with Feedback

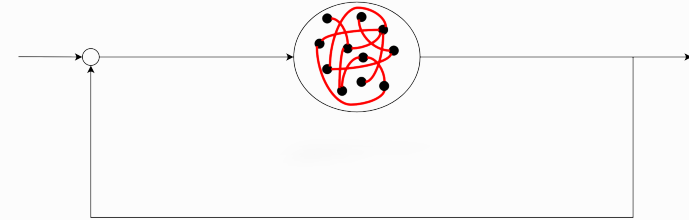
~~No Learning rule for control network available~~

~~Open loop Control:~~

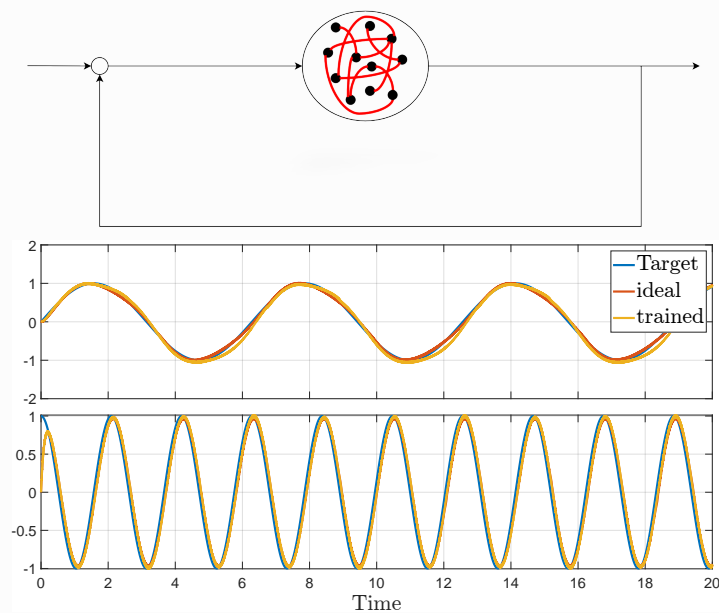
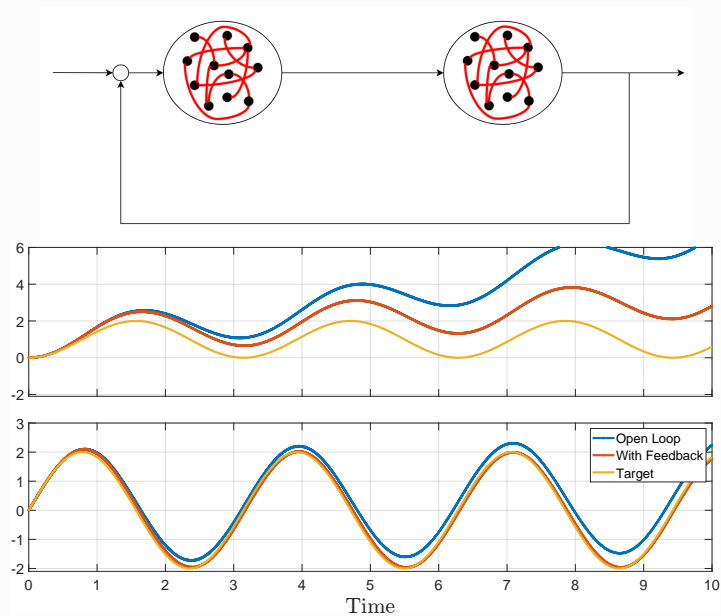
- ~~• Incapable of noise detection or correction~~
- ~~• No Compensation of Training error~~

~~Highly dependent on governing dynamics from~~ $\mathbf{c}_{\text{contr}} = \dot{\mathbf{x}}_{\text{ref}} - \mathbf{A}\mathbf{x}_{\text{ref}}$

~~Orthonormality restriction on Input Matrix~~ $\mathbf{B} \in \mathbb{B} := \{\mathbf{M} \mid \mathbf{M}\mathbf{M}^T = \mathbf{I}\}$



Examples





Conclusion

Conclusion

- Open loop and inaccurate learning of slow weights W^s need to be addressed.
 - Highly dependent on initial conditions in learning
 - Impressive accuracy
- In ideal conditions useable results achievable
 - Limited Applicability → Only of theoretical Interest
 - Results are somewhat translatable to NEF and LSMs

Choice between biologic plausibility or and Input Matrix Restriction for accurate results

Future Work

- Enable non-linear dynamics
- Obey Dale's Law for neuron excitation and inhibition
- Learning of En- and Decoder F
- Allow for synaptic delays

Bibliography I

- [BD15] Ralph Bourdoukan and Sophie Denève. “Enforcing balance allows local supervised learning in spiking recurrent networks”. In: **Advances in Neural Information Processing Systems**. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf.
- [BMD13] Martin Boerlin, Christian K. Machens, and Sophie Denève. “Predictive Coding of Dynamical Variables in Balanced Spiking Networks”. In: **PLOS Computational Biology** 9.11 (Nov. 14, 2013). Publisher: Public Library of Science, e1003258. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1003258](https://doi.org/10.1371/journal.pcbi.1003258). URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003258> (visited on 09/20/2022).

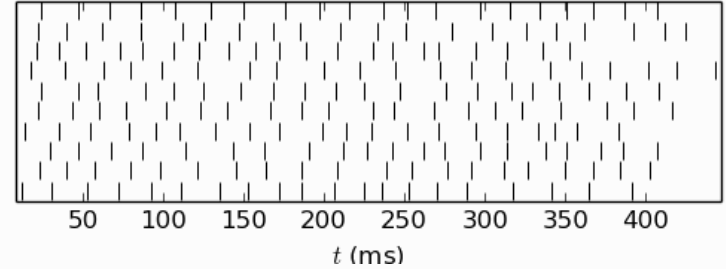
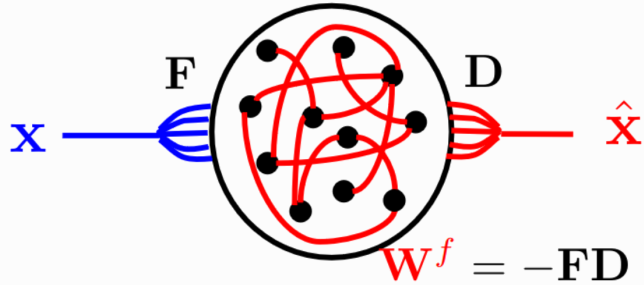
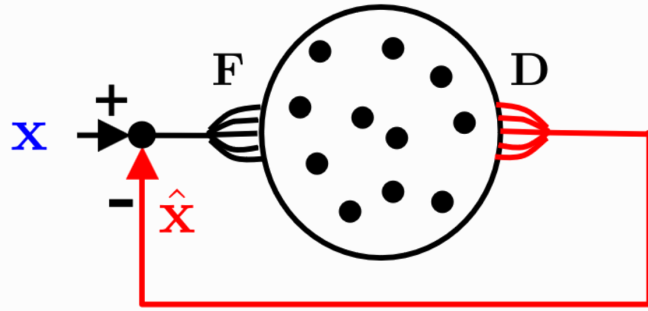
Bibliography II

- [HC19] Fuqiang Huang and ShiNung Ching. "Spiking networks as efficient distributed controllers". In: **Biological Cybernetics** 113.1 (Apr. 2019), pp. 179–190. ISSN: 0340-1200, 1432-0770. DOI: [10.1007/s00422-018-0769-7](https://doi.org/10.1007/s00422-018-0769-7). URL: <http://link.springer.com/10.1007/s00422-018-0769-7> (visited on 10/23/2022).
- [Xue+22] Xiaohe Xue, Ralf D. Wimmer, Michael M. Halassa, and Zhe Sage Chen. "Spiking Recurrent Neural Networks Represent Task-Relevant Neural Sequences in Rule-Dependent Computation". In: **Cognitive Computation** 15.4 (Feb. 2022), pp. 1167–1189. ISSN: 1866-9964. DOI: [10.1007/s12559-022-09994-2](https://doi.org/10.1007/s12559-022-09994-2). URL: <http://dx.doi.org/10.1007/s12559-022-09994-2>.



BackupSlides

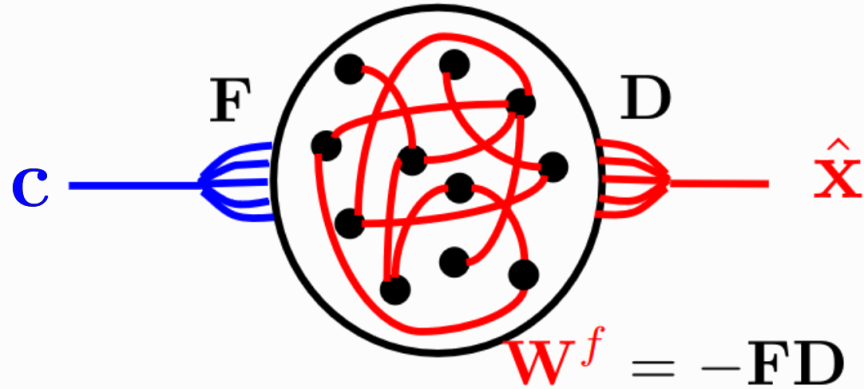
Autoencoder



$$\begin{aligned}\hat{x} &= Do(t) \\ \dot{r} &= -\lambda r + o(t)\end{aligned}\tag{2}$$

$$\dot{r} = -\lambda r + \sigma(t)$$

Autoencoder II

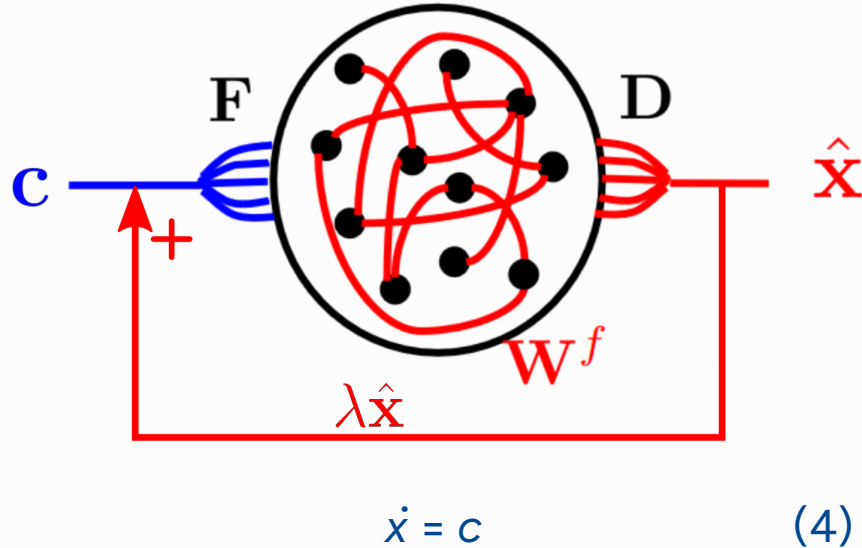


$$\begin{aligned} \dot{x} &= -\lambda x + c \\ \hat{x} &= Dx \end{aligned} \quad (3)$$

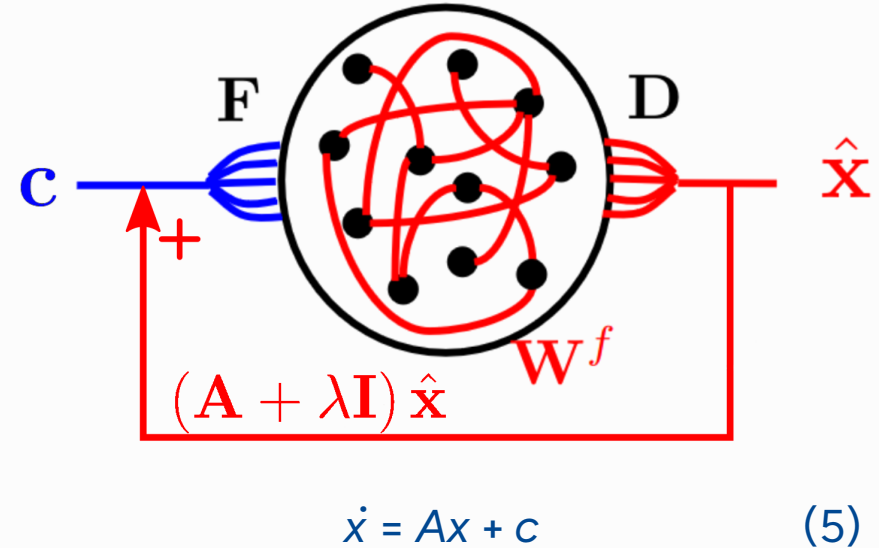
$$\dot{r} = -\lambda r + o(t)$$

$$\hat{x} = Dr$$

Autoencoder III

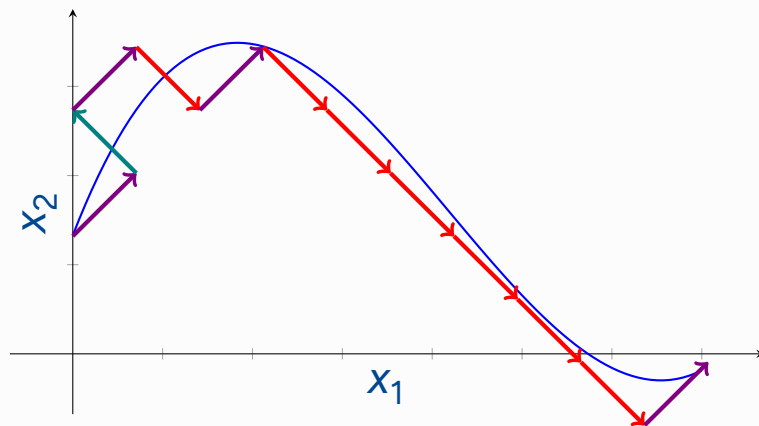
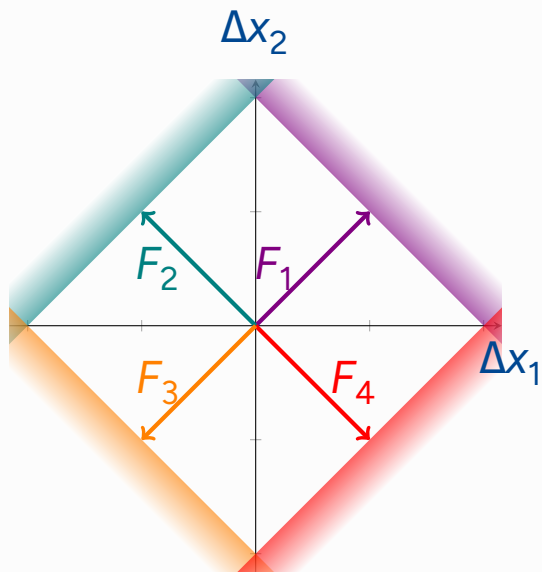


(4)

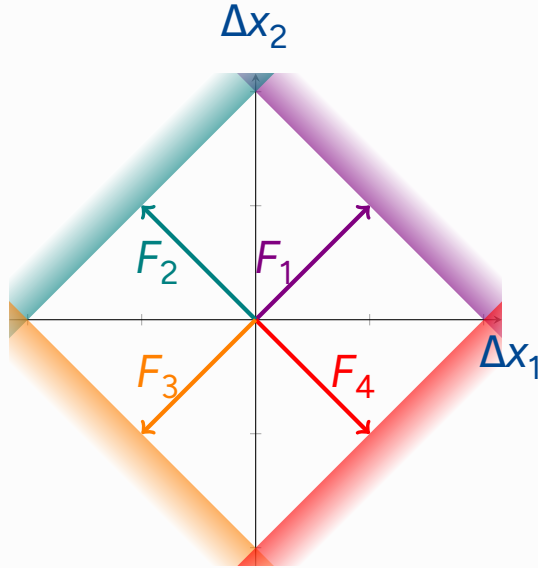


(5)

Geometric



Geometric



Minimize the cost J (Greedy)

$$J = \int_0^T \|x - \hat{x}\|_2^2 + C(r) dt \quad (6)$$

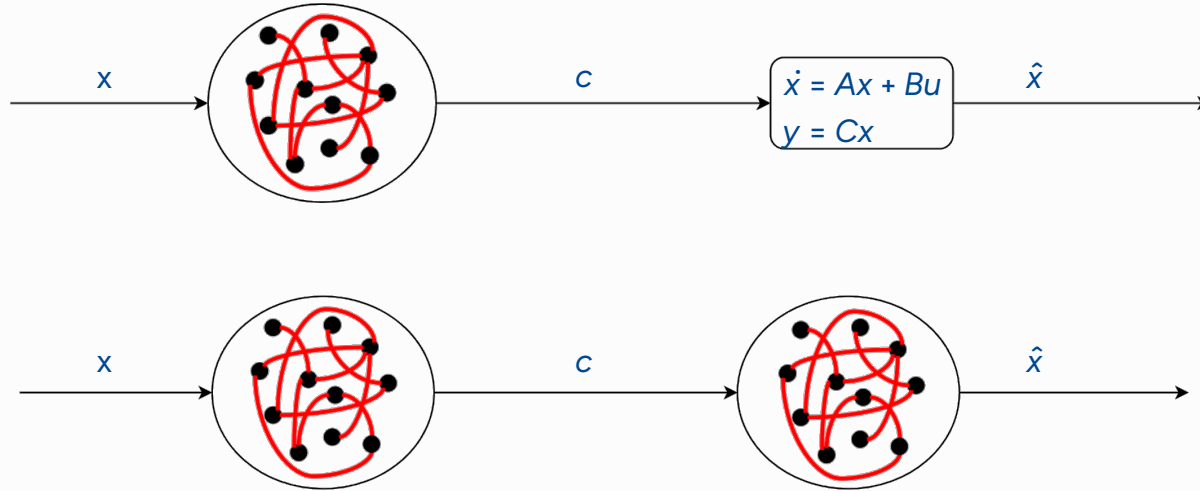
$$V_i = F_i(x - \hat{x}) - \mu r_i$$

$$\dot{V}_i = -\lambda_V V_i + F_i c(t) + W^f o(t) + W^s r(t) + \sigma_V \eta(t) \quad (7)$$

$$W^f = FF^T + \mu I$$

$$W^s = F(A + \lambda_d I)F^T$$

Control Concept



[HC19]

Control with SNN

$$u = F^T r + \Omega o(t) \quad (8)$$

Slow and Instantaneous decoding

$$\begin{aligned} \dot{V}(t) = & -\lambda_V V(t) + \Omega^T B^T A e(t) + \Omega^T B^T c(t) \\ & + W^s r(t) + W^f o(t) + \sigma_v \eta(t) \end{aligned} \quad (9)$$

Requires full state information on x
and \hat{x}

$$c = \dot{x} - Ax \quad (10)$$

It is necessary on $B \in \mathbb{R}^{n \times p}$

$$\text{rank}(B^T C^T) = p \quad (11)$$



Conclusion

Conclusion

- Acceptable results in ideal conditions

Conclusion

- Acceptable results in ideal conditions
- Rank condition is limiting factor

Conclusion

- Acceptable results in ideal conditions
- Rank condition is limiting factor
- Network noise is invisible to the control

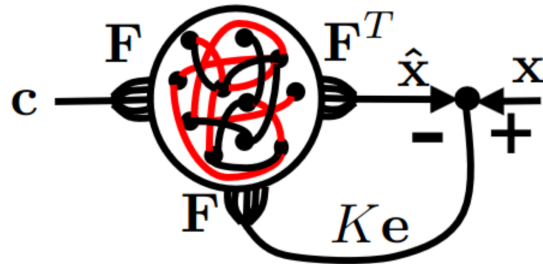
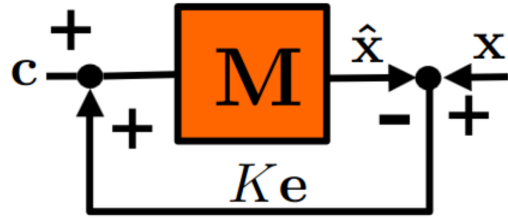
Conclusion

- Acceptable results in ideal conditions
- Rank condition is limiting factor
- Network noise is invisible to the control
- Simple open loop controller in the definition of c

$$\dot{\hat{x}} = (M - K\mathbf{I})\hat{x} + c + Kx$$

$$W^s = F(A + \lambda_d \mathbf{I})F^T$$

Slow Learning rule



[BD15]

Online Teacher-Student Scheme for M under $\dot{x} = Mx + c$

Matrix update under squared loss

$$\delta M \propto e\hat{x}^T \longrightarrow \delta W^s \propto F^T (e\hat{x}^T) F \approx F^T e r$$

(12)

Conclusion

- Very limited applicability
 - Open loop + rank condition limiting factor
 - Too inaccurate learning of slow weights W^s
 - Too dependent on initial conditions in learning
- In ideal conditions useable results achievable
 - Only of theoretical interest
 - Impressive accuracy
 - Results are somewhat translatable to NEF and LSMs

