Spiking Neural Networks for Control

Max Schaufelberger
February 7, 2024 — KTH Royal Institute of Technology

Table of Contents

Introduction

Simulation

Control

Learning

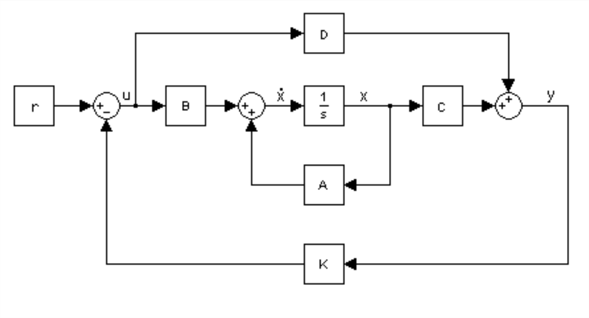
Learned Control

Conclusion

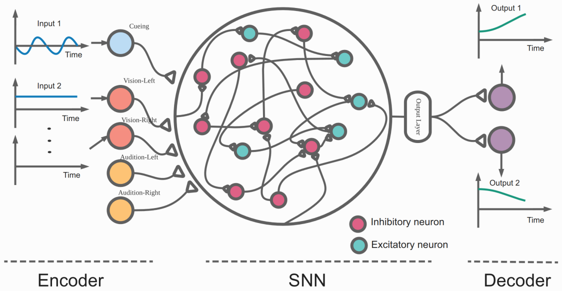
Introduction

What are we talking about

Control a Linear system



Use Spiking neural networks



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

Use Spiking neural networks

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

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”

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

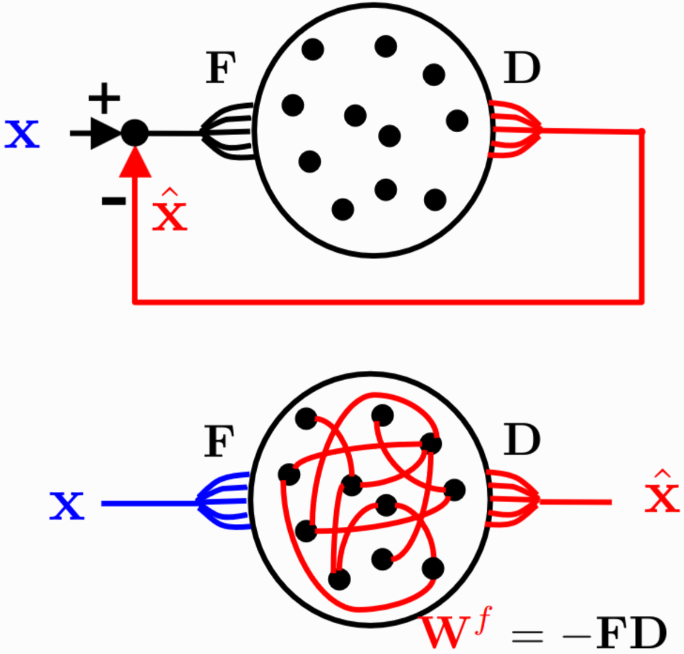
Apply biologically plausible learning rules to our network

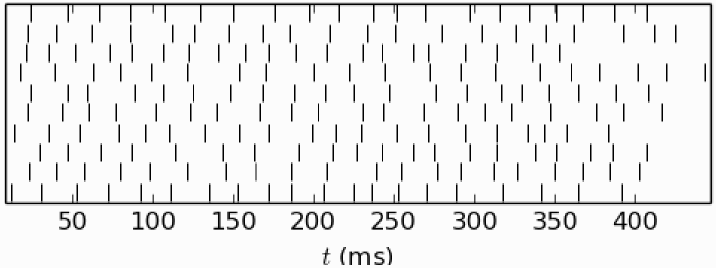
4. Combine

Integrate all three steps into a single controller

Simulation

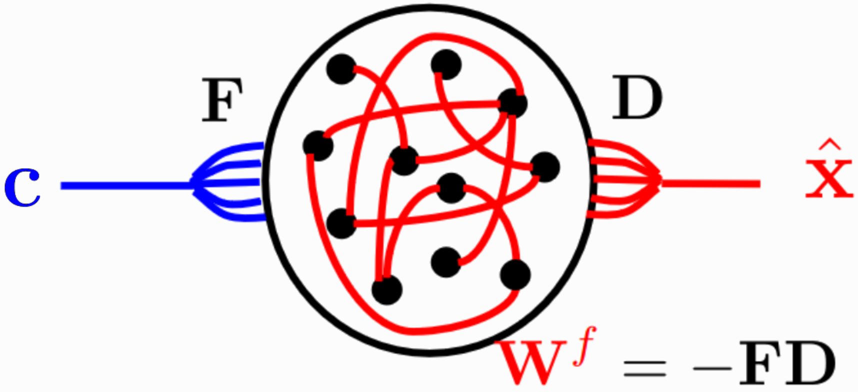
Autoencoder





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

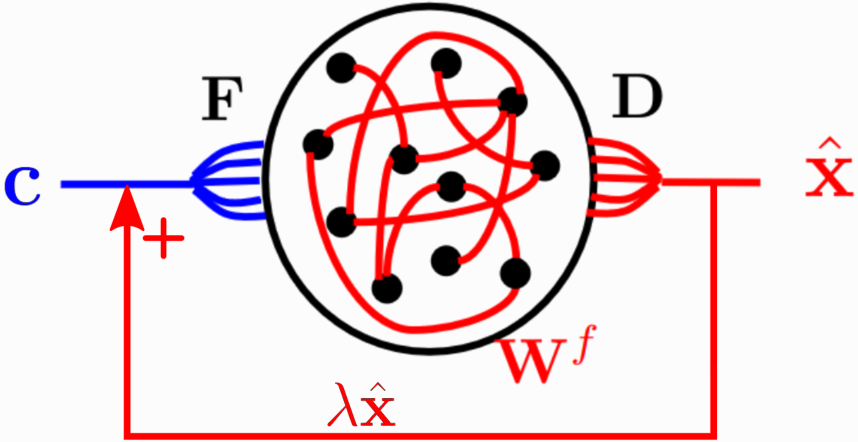
Autoencoder II



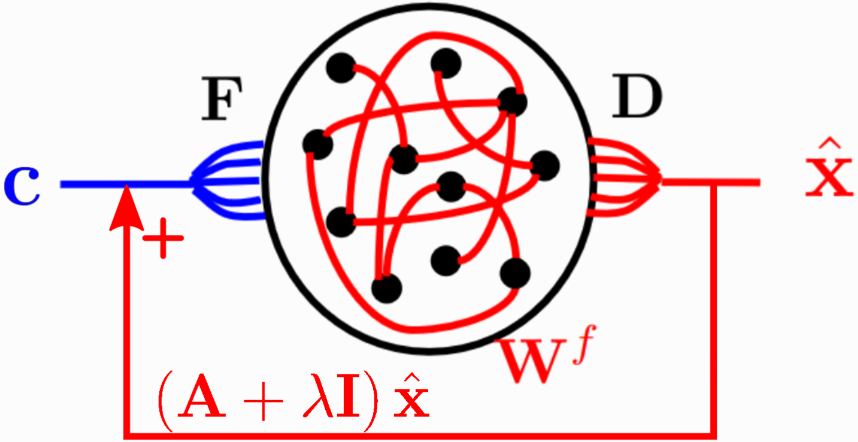
$$\begin{aligned} \dot{x} &= -\lambda x + c \\ \hat{x} &= Dr \end{aligned} \qquad (3)$$

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

Autoencoder III

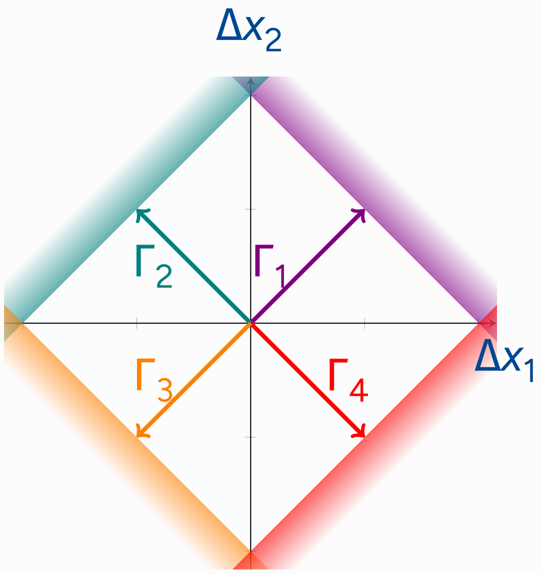


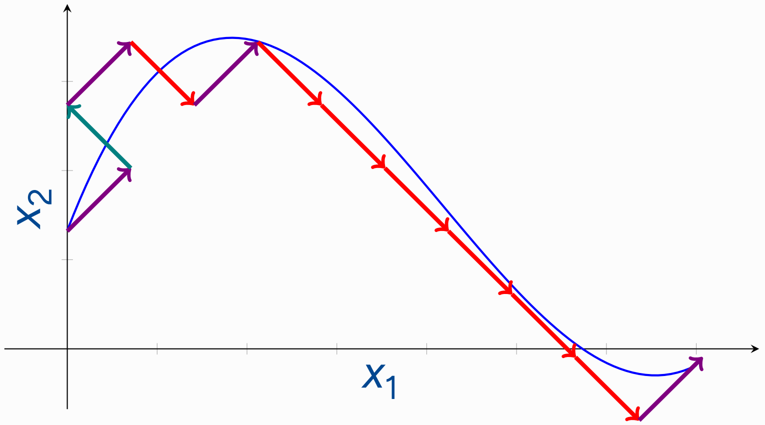
$$\dot{x} = c \tag{4}$$



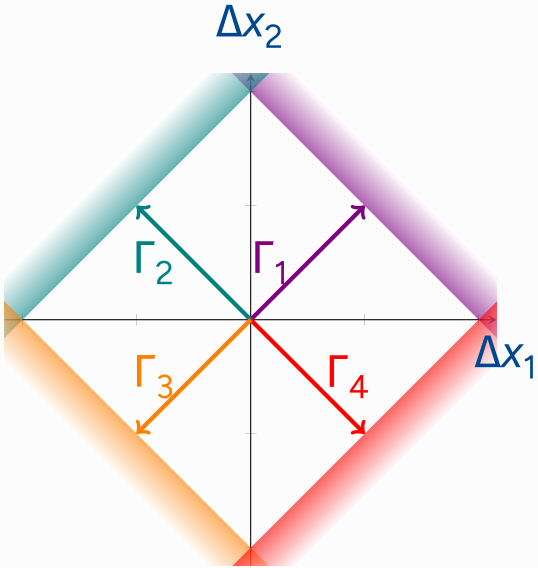
$$\dot{x} = Ax + c \tag{5}$$

Geometric





Geometric



Minimize the cost J (Greedy)

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

$$\begin{aligned} V_i &= \Gamma_i^T (x - \hat{x}) - \mu r_i \\ \dot{V}_i &= -\lambda_V V_i + \Gamma^T c(t) \\ &\quad + W^f o(t) + W^s r(t) + \sigma_V \eta(t) \\ W^f &= \Gamma^T \Gamma + \mu I \\ W^s &= \Gamma^T (A + \lambda_d I)\, \Gamma \end{aligned} \tag{7}$$

Example Simple

content...

Example Big

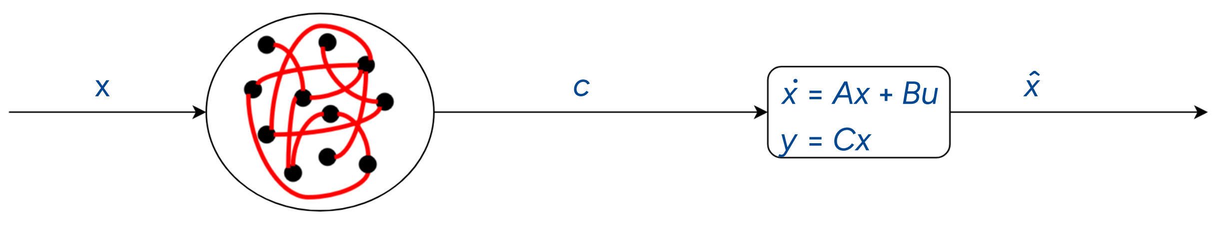
content...

Conclusion

content...

Control

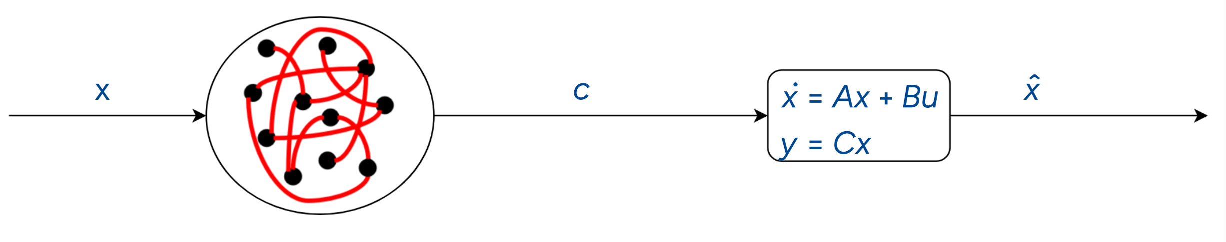
Control Concept



[HC19]

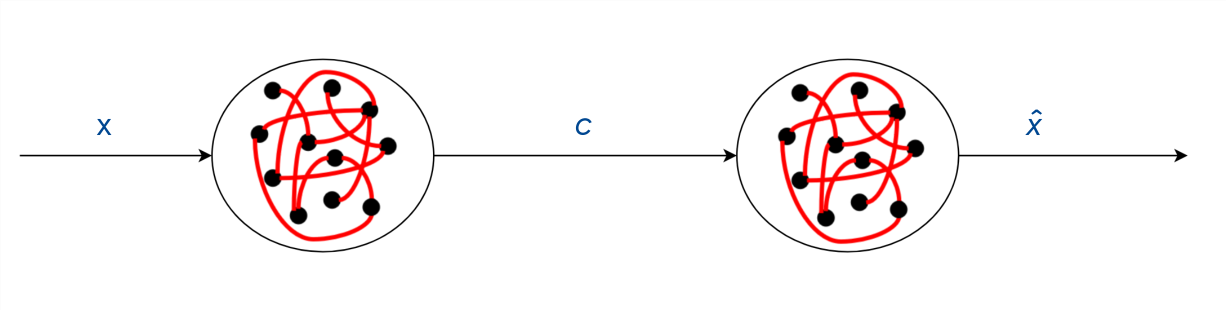
Add a separator here

Control Concept



[HC19]

Add a separator here



Control with SNN

$$u = \Gamma r + \Omega o(t) \tag{8}$$

Slow and Instantaneous decoding

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

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

$$c = \dot{x} - Ax \tag{10}$$

It is necessary on $B \in \mathbb{R}^{nxp}$

$$\operatorname{rank}(B^T C^T) = p \tag{11}$$

Example in Ideal Conditions

works fine+ add plot

Example with 2 networks

works bad+ add plot

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

Learning

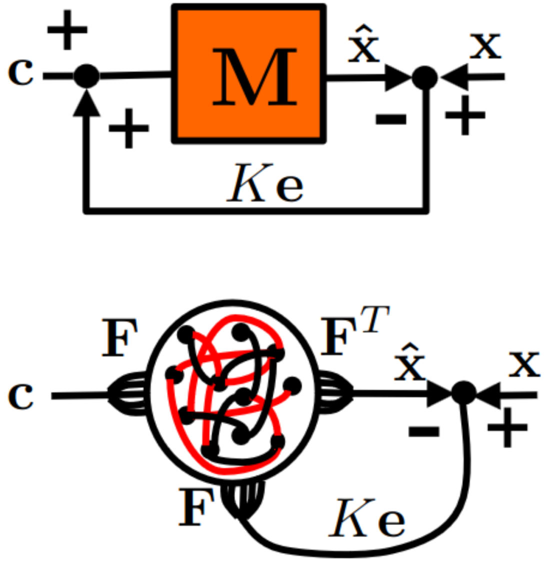
Fast Learning rule

Slow Learning rule
Online Teacher-Student Scheme

Fast Learning rule
content...

$$\dot{\hat{x}} = (M - K\mathbf{I})\hat{x} + c + Kx$$
$$W^s = \Gamma^T (A + \lambda_d \mathbf{I}) \Gamma$$

Slow Learning rule



[BD15]

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

$$\delta M \propto e\hat{x}^T \longrightarrow \delta W^s \propto \Gamma\left(e\hat{x}^T\right)\Gamma^T \approx \Gamma e r \tag{12}$$

replace the F with Γ in the picture!

Learned Control

Conclusion

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

Future Work

Future Work

- Enable non-linear dynamics

Future Work

- Enable non-linear dynamics
- Obey Dale’s Law for neuron excitation and inhibition

Future Work

- Enable non-linear dynamics
- Obey Dale’s Law for neuron excitation and inhibition
- Optimize Control

Future Work

- Enable non-linear dynamics
- Obey Dale’s Law for neuron excitation and inhibition
- Optimize Control
- Learning of En- and Decoder Γ

Future Work

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

Frame title

Block

Lorem ipsum!

Bibliography

[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`.

[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`. URL: `http://link.springer.com/10.1007/s00422-018-0769-7` (visited on 10/23/2022).

KTH
VETENSKAP
OCH KONST