Machine learning to efficiently linearize system for control

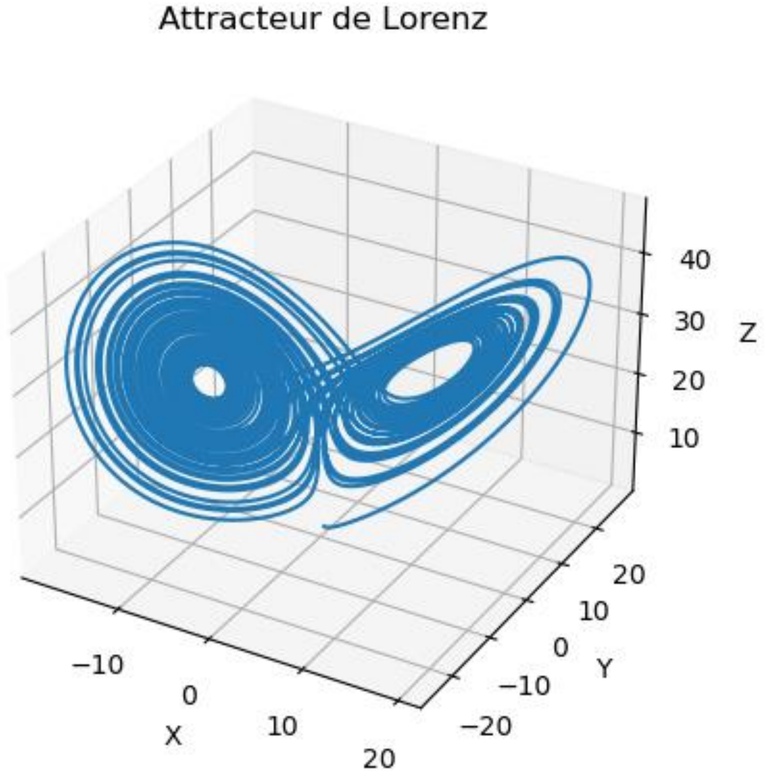
Outline

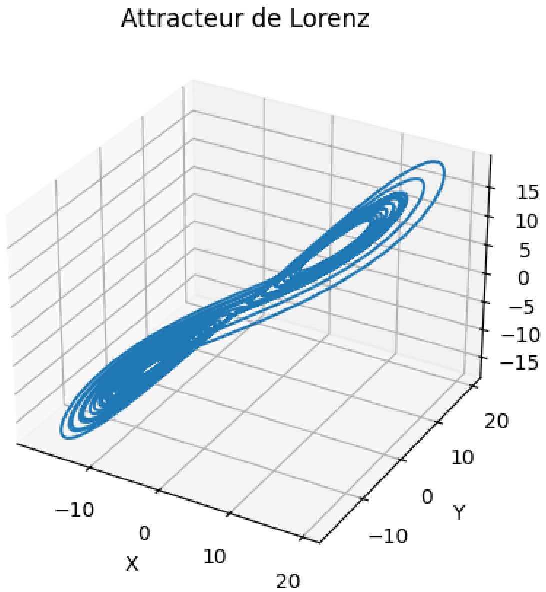
- Partial observation
- RSNN + framework
- Efficiency

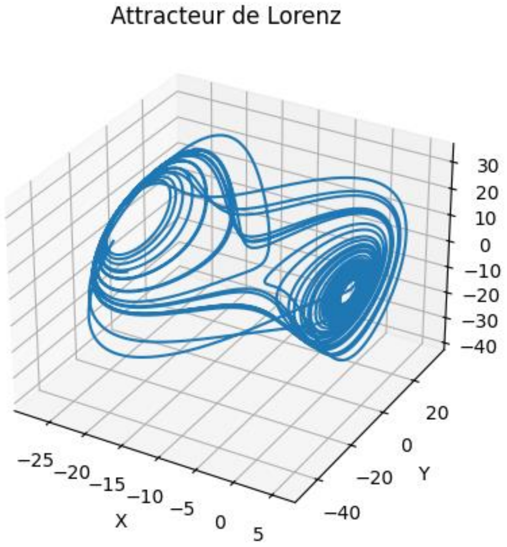
Partial observation

- System 3 dimension -> observe 1st dimension
- Exemple : lorentz / $\dot{x} = -\lambda\, x + \mathrm{W}\tanh(x)$

Observation



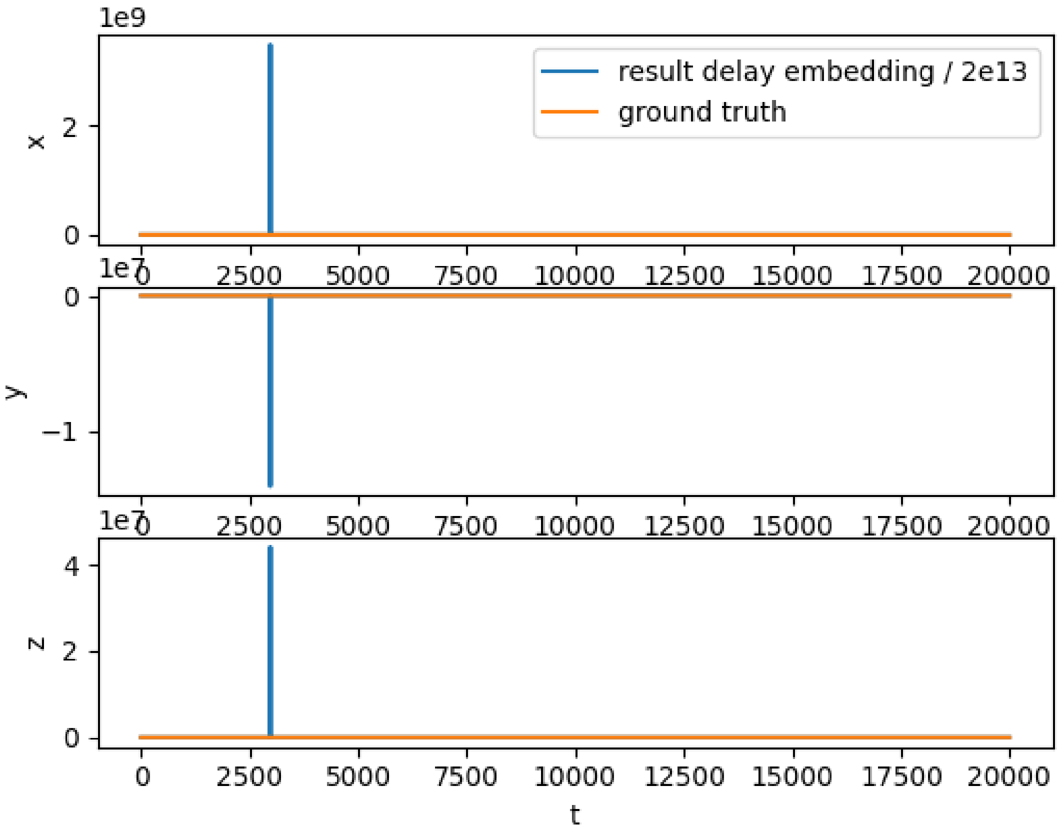




- Suppose the dimensional state vector xt evolves according to an unknown but continuous and (crucially) deterministic dynamic. Suppose, too, that the one-dimensional observable y is a smooth function of x, and **“coupled”** to all the components of x. Now at any time we can look not just at the present measurement y(t), but also at observations made at times removed from us by multiples of some lag τ : yt−τ , yt−2τ , etc. If we use k lags, we have a k-dimensional vector. One might expect that, as the number of lags is increased, the motion in the lagged space will become more and more predictable, and perhaps in the limit k → ∞ would become deterministic. In fact, the dynamics of the lagged vectors become deterministic at a finite dimension; not only that, but the deterministic dynamics are completely equivalent to those of the original state space! (More exactly, they are related by a smooth, invertible change of coordinates, or diffeomorphism.) The magic embedding dimension k is at most 2d + 1, and often less.

Shalizi, Cosma R. (2006). "Methods and Techniques of Complex Systems Science: An Overview"

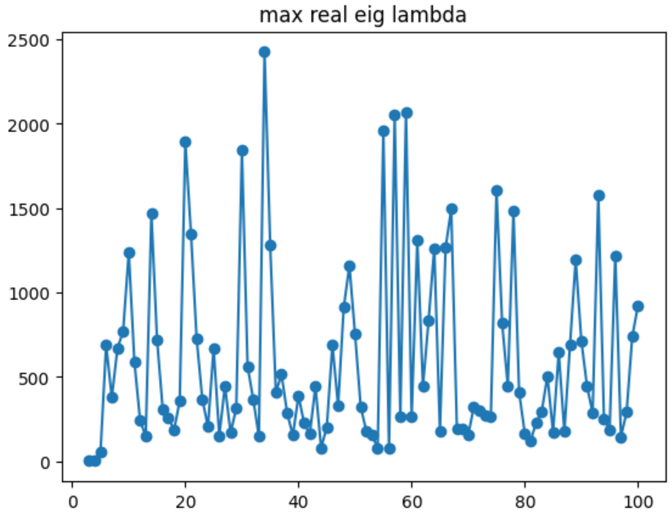
Results

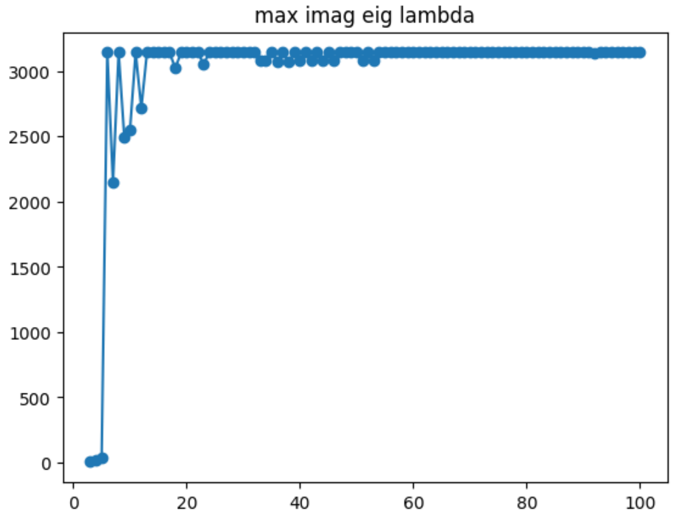


- Problem -> explode

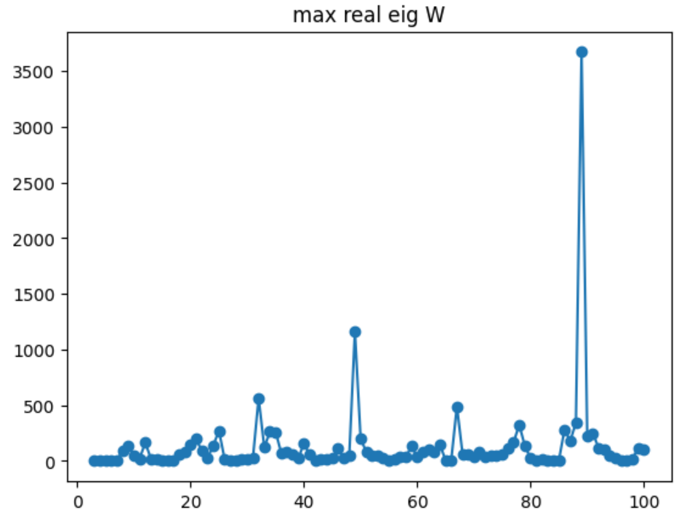
Test dimension

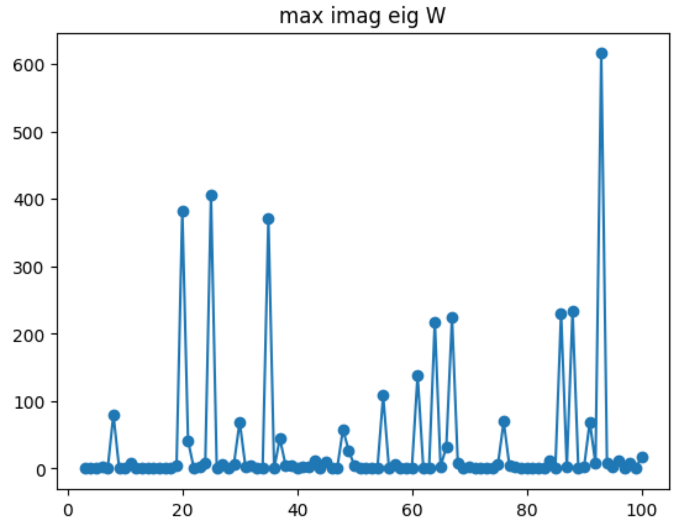
$$\dot{x} = -\lambda\, x + \mathrm{W}\tanh(x)$$



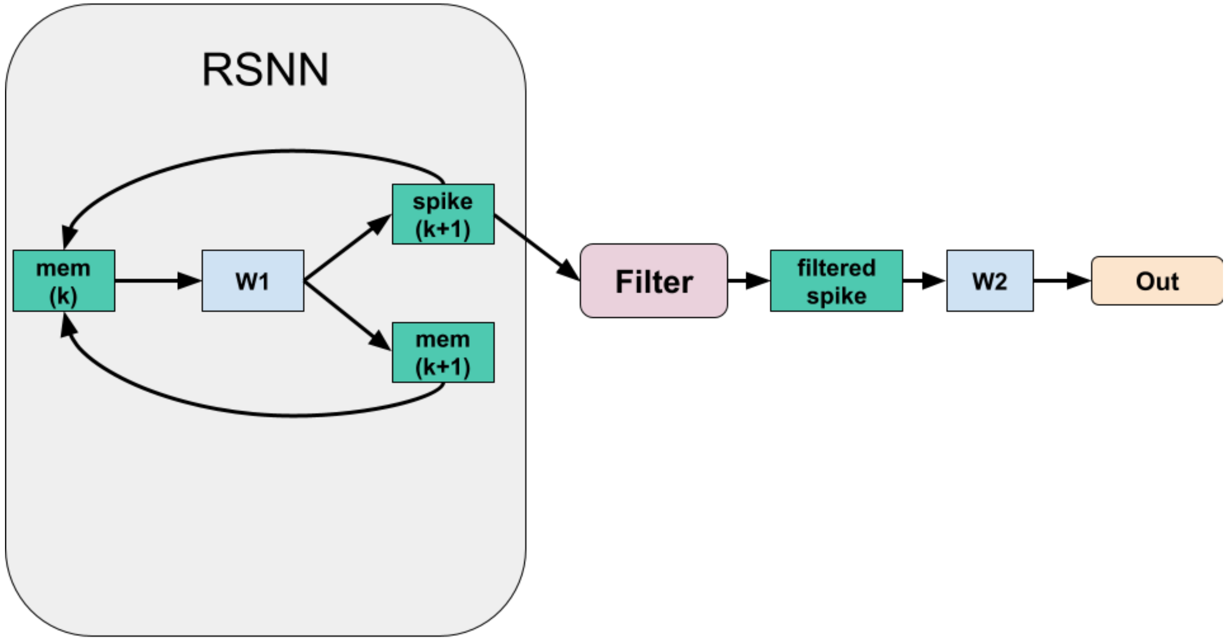


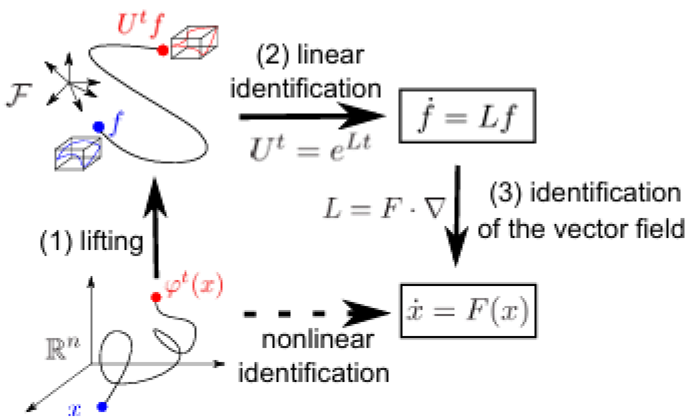
The linear part creates oscillations and explosions !!





RSNN + framework





Equations

- Membrane potential RSNN (1) $\dot{v}_1 = -\frac{v_1}{\tau_1} + W_1 f(v_1)$

- Membrane potential filtered spikes (2) $\dot{v}_2 = -\frac{v_2}{\tau_2} + f(v_1)$

f : spike

- Output (3) $x = W_2 v_2$

Equations transformation

- (2) and (3) => (4) $\dot{x} = -\frac{x}{\tau_2} + W_2 f(v_1)$

- (1) and (4) => (5) $\boxed{\dot{X} = -\frac{X}{\tau} + W f(X)}$

$\boxed{X = \begin{pmatrix} v_1 \\ x \end{pmatrix}}$ $\quad W = (W_1 \;\; W_2)$ $\quad f(X) = \begin{pmatrix} f(v_1) \\ f(v_1) \end{pmatrix}$

Problems

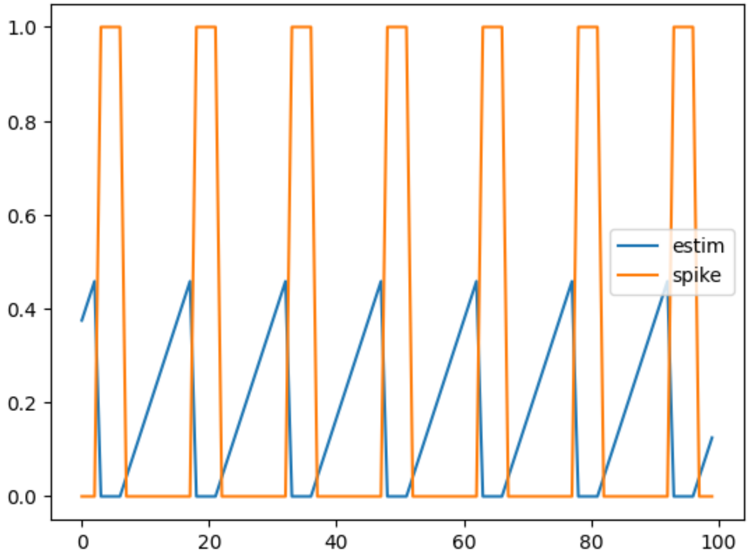
- Data ???　　　Need to give data for v_1　　　and dimension ?

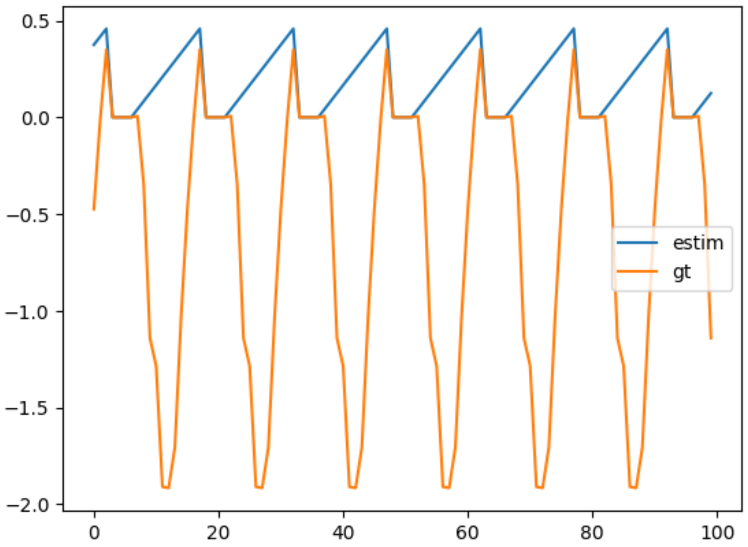
- Partial observation ??

Inversion SNN

- Finding v_1 from v_2

- $spike[t] = v_2[t] > \beta[v_2[t-1]$





Inversion SNN

- Finding v_2 from x

$$v_2 = W_2^{\dagger} x$$

Problems

- Equation are decoupled
- Potential only positive (really a problem? only spike matter)

Efficiency

- Efficiency in data size/sampling http://arxiv.org/abs/1709.02003v4 (lifting method)
- Efficiency in frequency learning On the Activation Function Dependence of the Spectral Bias of Neural Networks (2022)
- Efficiency in dimension
- Efficiency in learning (amount of computation) http://arxiv.org/abs/1709.02003v4 (lifting method)/Nonlinear computations in spiking neural networks through multiplicative synapses (2021)

- Efficiency in computation

- …

Questions ?

To do

- Control multiplicative synapse ?
- SNN learn Lorentz ?