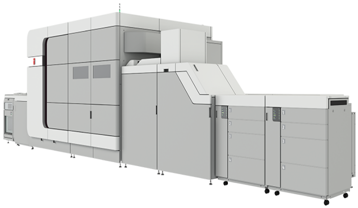
Keeping Feedback in Neural Networks

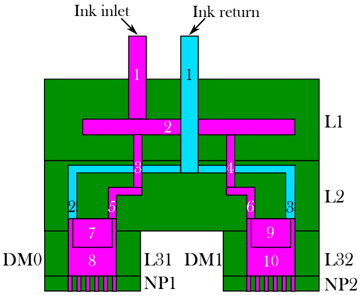
Amritam Das

Control Group, Department of Engineering

A Few Words About My PhD

Digital twin for thermo-fluidic processes in inkjet printing



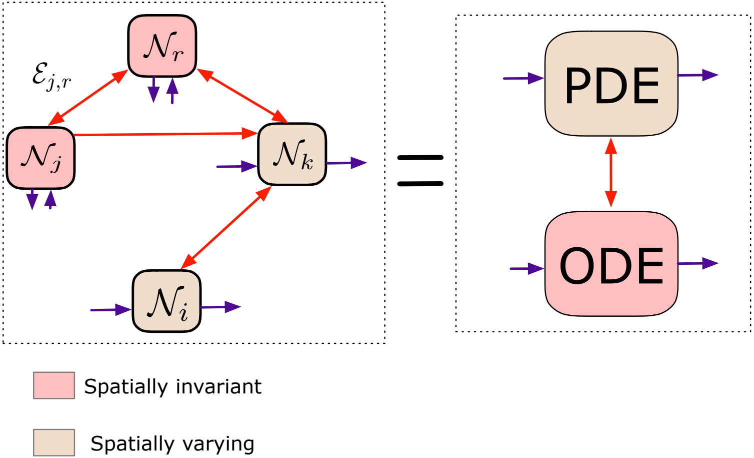


Requirements:

- How to achieve specific temperature at individual droplets?
- Is it possible to achieve that without adding new sensors or actuators?

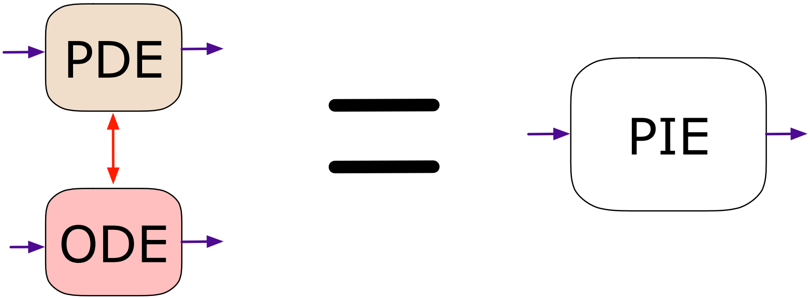
A Few Words About My PhD

Digital twin for thermo-fluidic processes in inkjet printing



A Few Words About My PhD

Digital twin for thermo-fluidic processes in inkjet printing



Advantages of Partial Integral Equation (PIE):

- Analysis and synthesis on PIE only requires LMIs (no numerical approximation)
- A software package **PIETOOLS** is co-developed to perform functionalities of PIE

*Das (2020): A Digital Twin for Controlling Thermo-Fluidic Processes

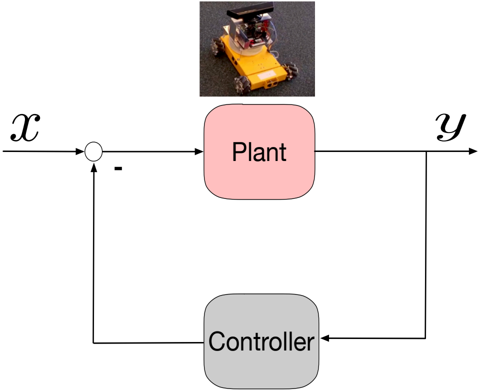
Today's Talk: Keeping Feedback in Neural Networks

Based on my research for past 8 months and on some of the ideas I believe worth pursuing

Outline

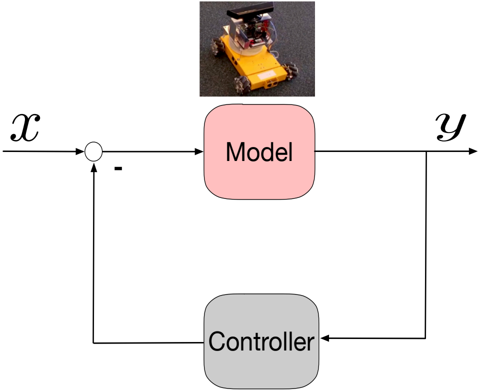
1. Feedback control
2. Intro to RNNs
3. View RNNs as feedback control systems
4. Connecting convex optimization and computation of RNNs
5. Some open problems
6. Concluding remarks

Shaping input's sensitivity at the output



What’s Feedback Control All About?

Shaping input’s sensitivity at the output



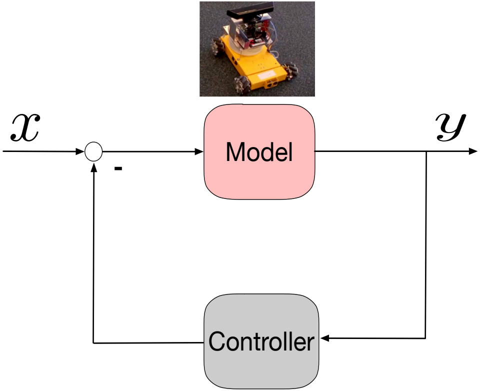
- "Ideally, a genuine theory of feedback is one that describes the fundamental properties of systems that belong in a given region of *objectives*, *ignorance* and *constraints*.."- I.M. Horowitz(1963)

Some merits in linear regime

- I/O properties have graphical (visual) interpretation: Nyquist theory
- Quantifiable (derivable) margins of stability and robustness directly from I/O rerlattion

What's Feedback Control All About?

Shaping input's sensitivity at the output



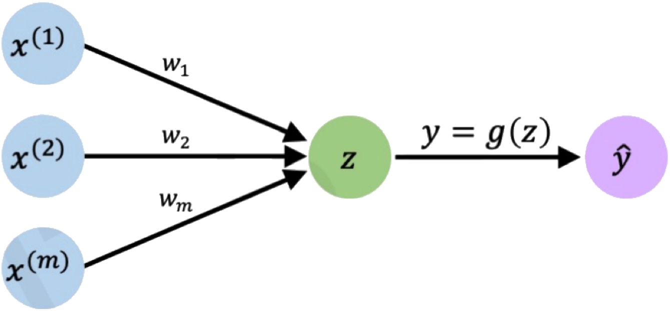
- "Ideally, a genuine theory of feedback is one that describes the fundamental properties of systems that belong in a given region of *objectives*, *ignorance* and *constraints*.."- I.M. Horowitz(1963)

However...

- No direct extension towards nonlinear elements
- Some approximating result available, e.g. Circle criterion, Harmonic Analysis

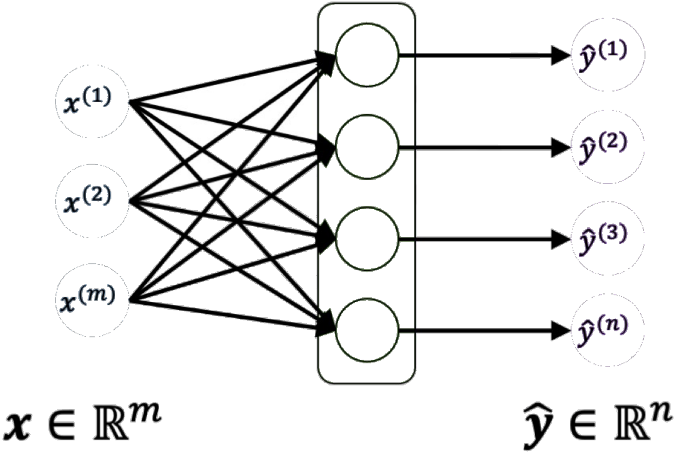
Example of Feedback in Nonlinear Systems: Recurrent Neural Network

Feedforward Perceptron: universal approximator of static plant



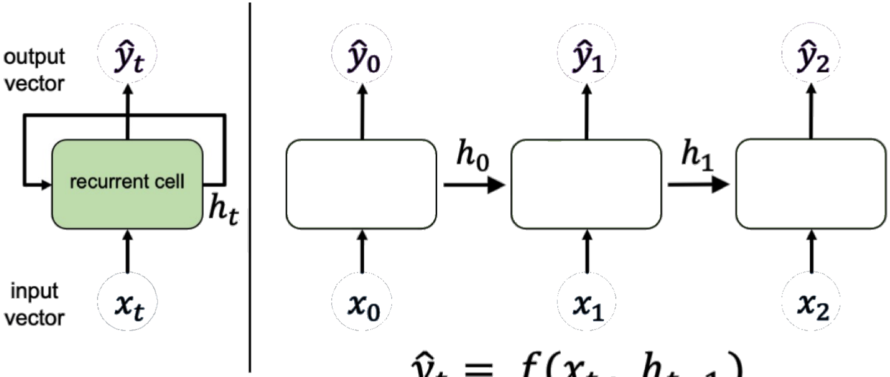
Example of Feedback in Nonlinear Systems: Recurrent Neural Network

Feedforward Perceptron: universal approximator of static plant



Neural Networks: Role of Recurrence

Recurrent Network: universal approximator of dynamic plant (involving sequential inputs and outputs)

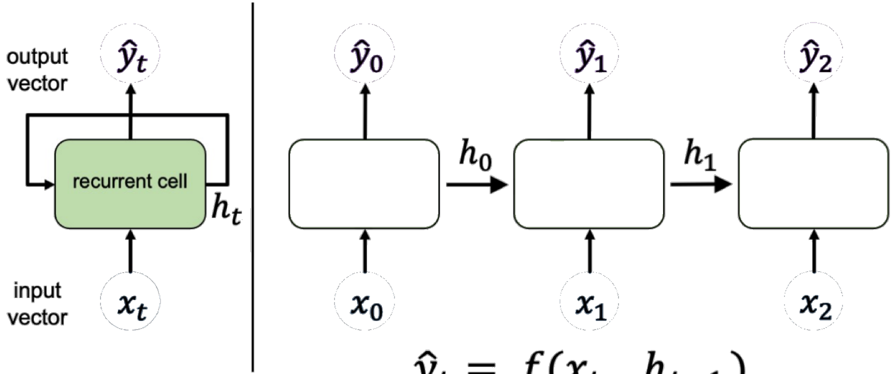


$$\hat{y}_t = f(x_t, h_{t-1})$$

output   input   past memory

Neural Networks: Role of Recurrence

Recurrent Network: universal approximator of dynamic plant (involving sequential inputs and outputs)



$$\underbrace{\hat{y}_t}_{\text{output}} = f(\underbrace{x_t}_{\text{input}}, \underbrace{h_{t-1}}_{\text{past memory}})$$

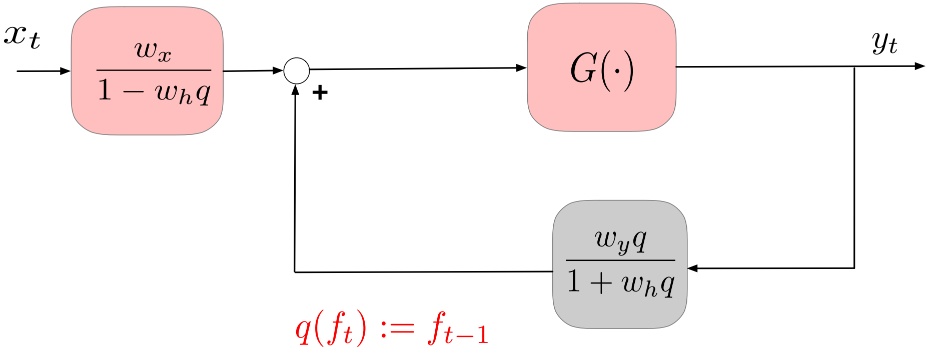
Feedback is inherent part of the structure!

Feedback Structure of RNNs: An Illustration

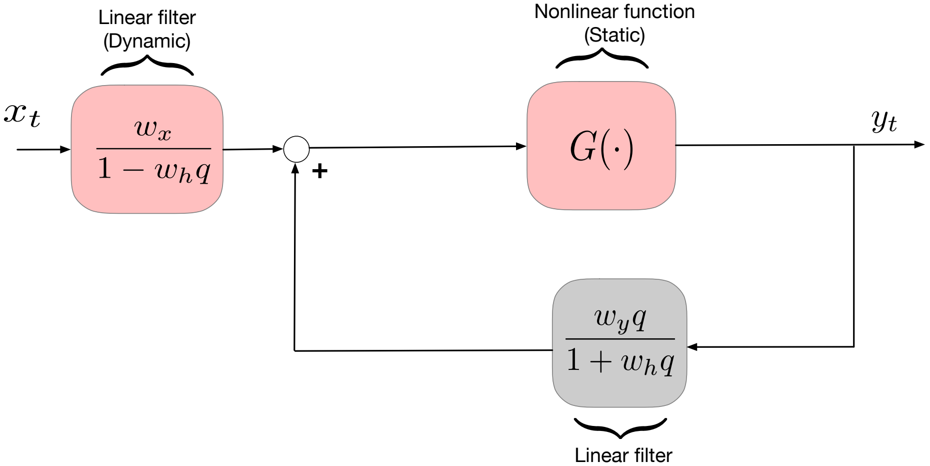
Canonical version of an RNN $:= \begin{cases} h_t = w_h h_{t-1} + w_y y_{t-1} + w_x x_t \\ y_t = G(h_t) \end{cases}$

Feedback Structure of RNNs: An Illustration

Canonical version of an RNN $:= \begin{cases} h_t = w_h h_{t-1} + w_y y_{t-1} + w_x x_t \\ y_t = G(h_t) \end{cases}$



Feedback Structure of RNNs: An Illustration

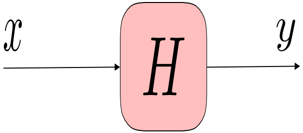


I/O relation of the pre-trained RNN

$$0 = \underbrace{\frac{w_y q}{1 + w_h q} y_t}_{\text{linear}} + \underbrace{\frac{w_x}{1 - w_h q} x_t}_{\text{shift}} - \underbrace{G^{-1} y_t}_{\text{static nonlinear}}$$

Control Theoretic Interpretation of the Feedback Structure

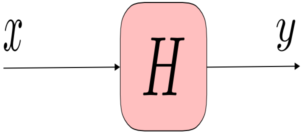
- Let us look at one element:



- The supplied energy (in incremental sense) $\Delta V := \left\langle x_1 - x_2, H(x_1) - H(x_2) \right\rangle$
- H is incrementally passive if $\Delta V \geq 0$

Control Theoretic Interpretation of the Feedback Structure

- Let us look at one element:



- The supplied energy (in incremental sense) $\Delta V := \left\langle x_1 - x_2, H(x_1) - H(x_2) \right\rangle$
- H is incrementally passive if $\Delta V \geq 0$

In the language of operator theory

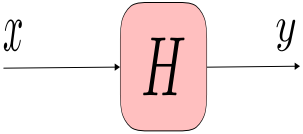
Incremental passivity is synonymous to *Monotonicity* on signal space

(A positive change in the input should cause a positive change in the output)

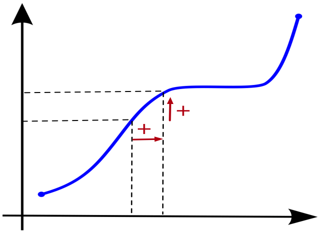
*Camlibel and Schumacher (2016), "Linear passive systems and maximal monotone mappings"

Control Theoretic Interpretation of the Feedback Structure

- Let us look at one element:



- The supplied energy (in incremental sense)
  $\Delta V := \left\langle x_1 - x_2, H(x_1) - H(x_2) \right\rangle$
- H is incrementally passive if $\Delta V \geq 0$



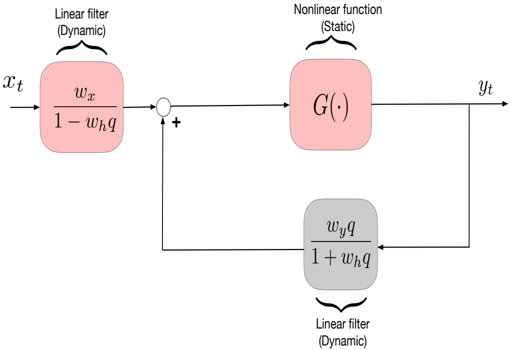
In the language of operator theory

Incremental passivity is synonymous to *Monotonicity* on signal space

(A positive change in the input should cause a positive change in the output)

*Camlibel and Schumacher (2016), "Linear passive systems and maximal monotone mappings"

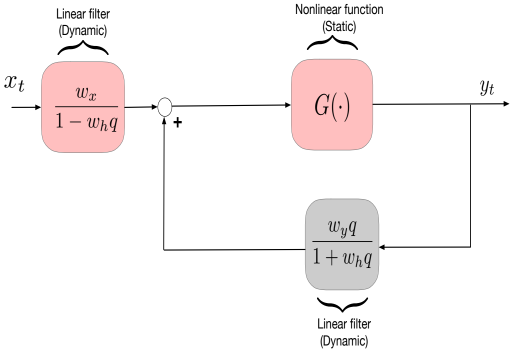
Feedback Structure of RNNs: Passivity/Monotonicity View-Point



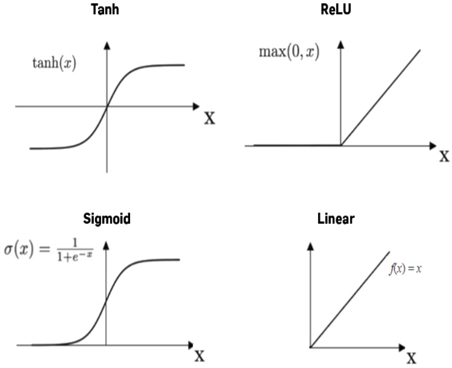
I/O relation of the pre-trained RNN

$$0 = \underbrace{\frac{w_y q}{1 + w_h q}}_{\text{linear}} y_t + \underbrace{\frac{w_x}{1 - w_h q}}_{\text{shift}} x_t - \underbrace{G^{-1} y_t}_{\text{static nonlinear}}$$

Feedback Structure of RNNs: Passivity/Monotonicity View-Point



$$0 = \underbrace{\frac{w_y q}{1+w_h q} y_t + \frac{w_x}{1-w_h q} x_t}_{\text{Montone}} - \underbrace{G^{-1} y_t}_{\text{Montone}}$$



- *Characterize each operators based on monotonicity*

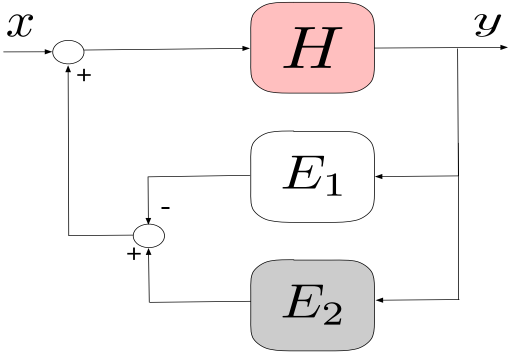
  Operator F is called monotone iff

  $\left\langle z - r, F(z) - F(r) \right\rangle \geq 0$ for any inputs $z, r \in \mathcal{X}$

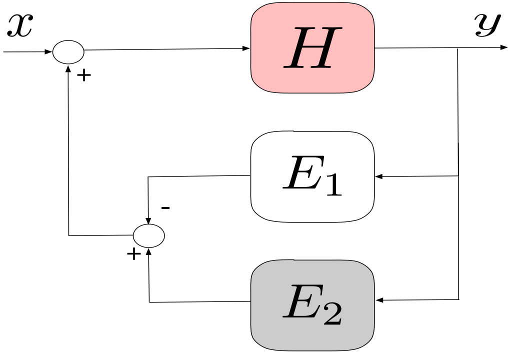
I/O relation of the entire RNN is a difference of monotone relation

Note: In case of -ve feedback, i/o relationship would be monotone. Classical feedback control is all about that

Feedback Structure of RNNs: General Setting



Feedback Structure of RNNs: General Setting



- H, E_1, E_2 are monotone operators
- I/O relation: $0 = H^{-1}(y) + E_1(y) - E_2(y) - x$
- Covers many classes of RNNs, e.g. LSTM, GRU

Main Question

How to develop a scalable (algorithmic) tool for RNNs?

e.g. Given $x = x^*$, find $y \in \mathcal{Y}$ such that $0 = H^{-1}(y) + E_1(y) - x^* - E_2(y)$

Computational Framework: Connecting Monotonicity and Convexity

G is a monotone operator iff $\left\langle z - r, G(z) - G(r) \right\rangle \geq 0$ for any inputs $z, r \in \mathcal{X}$

- **Key property:** The gradient of a convex function is monotone*
- **Consequence:** Minimizing a convex function is equivalent to finding the zero of a monotone operator

$$\textbf{If } F(x) \textbf{ is convex, } \arg\min_{x \in \mathbb{R}^n} F(x) \implies \text{find } x \in \mathbb{R}^n \ \ \textbf{such that} \ \ 0 = \nabla F(x)$$

- **Application:** Large-scale optimization, distributed optimization for multi-agent systems, distributed Nash equilibria for cooperative games**

---

* Rockafellar (1976): Monotone Operators and the Proximal Point Algorithm.
** Ryu & Lin (2020): Large-Scale Convex Optimization via Monotone Operators

Computational Framework: Finding Zero of Monotone Operators

Given a montone operator R, a solution to the following problem

$$0 \in R(x)$$

is a fixed point of the following relation

$$x_f = F(x_f), \quad F \text{ is derived from } R$$

You can find x_f by solving the fixed point iteration

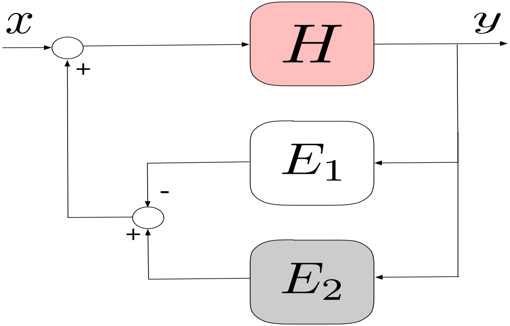
$$x_f^{k+1} = F(x_f^k)$$

If $F := (I + \alpha R)^{-1}$ (a.k.a resolvent), then the fixed point iteration converges. $\alpha > 0$ is the step-size

Advantages

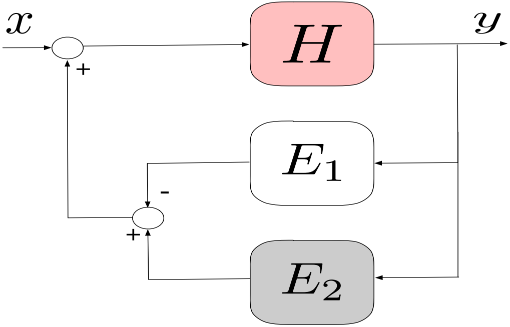
- Finding resolvents for linear operator and activation functions are easy
- Solving $0 \in \sum_{i=1}^{n} R_i(x)$ can be made scalable by splitting the computation of each resolvent F_i

Finding Zero of Difference between Monotone Operators



$$0 \in H^{-1}(y) + E_1(y) - x^* - E_2(y)$$

Finding Zero of Difference between Monotone Operators



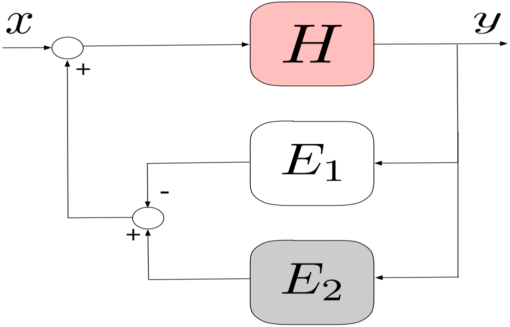
$$0 \in H^{-1}(y) + E_1(y) - x^* - E_2(y)$$

- Inspiration from Difference of Convex optimization:

  If $F(x), G(x)$ are convex,

  critical pt. $F(x) - G(x) \implies 0 \in \nabla F(x) - \nabla G(x)$

Finding Zero of Difference between Monotone Operators



$$0 \in H^{-1}(y) + E_1(y) - x^* - E_2(y)$$

- Inspiration from Difference of Convex optimization:

  If $F(x), G(x)$ are convex,

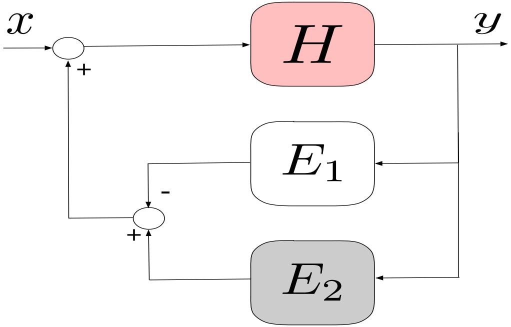
  critical pt. $F(x) - G(x) \implies 0 \in \nabla F(x) - \nabla G(x)$

- **Algorithm:**

  1: **Repeat** $x^* = \arg\min_x F(x) - \underbrace{\hat{G}(x)}_{\text{Linearized } G(x)}$

  2: **until** stopping criterion is satisfied

Finding Zero of Difference between Monotone Operators



$$0 \in \underbrace{H^{-1}(y) + E_1(y) - x^*}_{A(y)} \underbrace{- E_2(y)}_{B(y)}$$

- **Algorithm:**

1: Given: An initial guess y_0
2: **Repeat**
3: **Do**

$$y_{i+1} = \operatorname*{find}_{y} 0 \in A(y) - B(y_i)$$

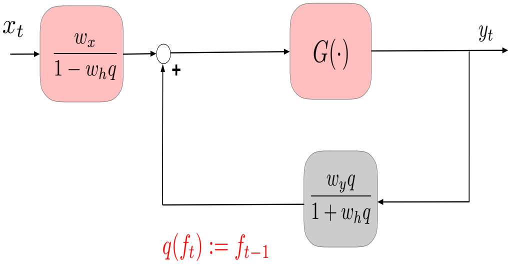
$$i := i + 1$$

4: **until** stopping criterion is satisfied

Results in a convergent and scalable algorithm

*Das, Chaffey, and Sepulchre (2021): Oscillations in Mixed-Feedback Systems, CDC'21(under review, Journal in progress)
*Das (2021): Optimization Tools for LSTMs, NIPS'21 (to be submitted)

Two Interesting Problems: Robustness Against Adversarial Inputs

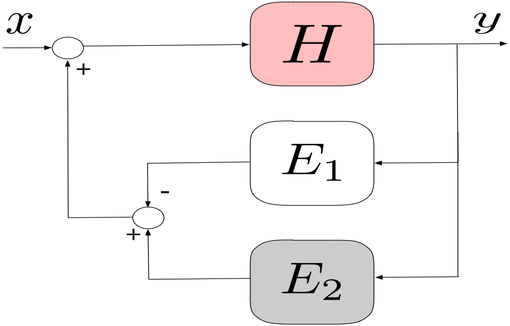


Question

- Determine the (convex) set $\mathcal{X} \ni x$ such that $\|y - y^*\| \leq \gamma$
- Given the (convex) set $\mathcal{X} \ni x$ find w_x, w_y, w_h and the minimum value of γ such that $\|y - y^*\| \leq \gamma$

Note: In control theory, this is called robust-control problem (uses convex optimization)

Two Interesting Problems: Adding Controller



$$0 = H^{-1}(y) + E_1(y) - x - E_2(y)$$

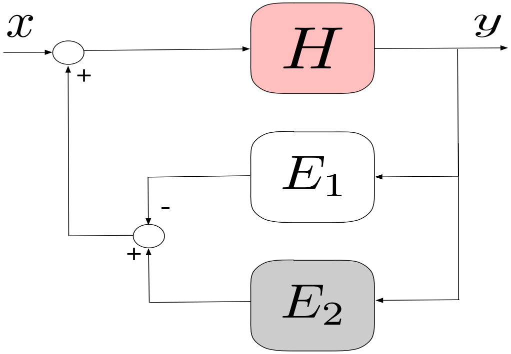
Question

- How to parameterize and learn E_1?

Note: In computational neurosience, this combination of $+$ve and -ve feedback is the source of spiking (the way neurons communicate)

*Burghi (2020): Feedback for Neuronal System Identification

RNNs are mix-signed feedback interconnection of monotone operators



$$0 = H^{-1}(y) + E_1(y) - x - E_2(y)$$

- Scalable to deep networks by splitting the computation
- H, E_1, E_2 may consist of more sophisticated operators, e.g. Laplacian, convolution etc.
- x, y can be continuous functions, even random variable (however, the algorithm changes)
- **Adding physical/dynamical constraints are possible** (e.g. PINNs)
- **Nyquist-like diagram is possible for monotone operartor** (It is called Scaled Relative Graph-SRGs)

Thank You!