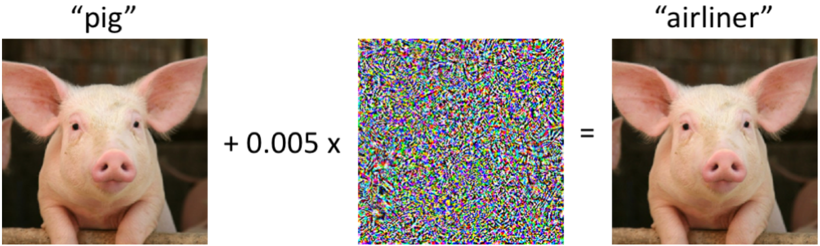
Robust Learning for Dynamics and Control via Contracting Neural Models

Ian R. Manchester,
Australian Centre for Robotics, University of Sydney
joint work with:
Ruigang (Ray) Wang, Max Revay, Nic Barbara



ACC Workshop 2023

Motivation



Small input perturbation $x + \Delta x$

$\Downarrow$

Large output change $y + \Delta y$

Image: Aleksander Madry, MIT.

(a) Image

(b) Prediction

(c) Adversarial Example

(d) Prediction

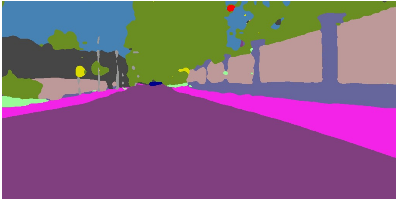


Image: Metzen et al, Universal Adversarial Perturbations Against Semantic Image Segmentation, 1704.05712v3.

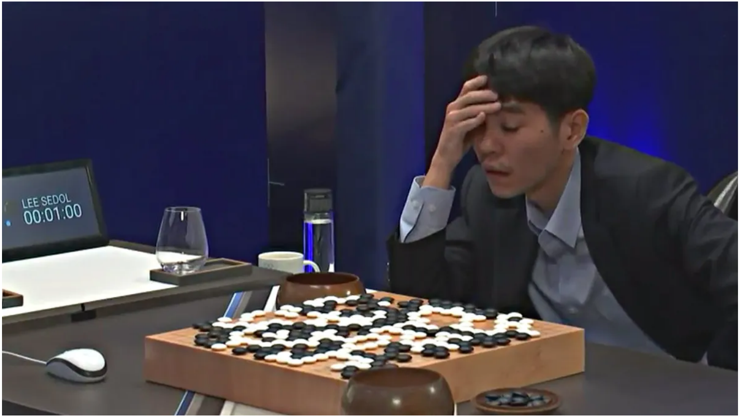
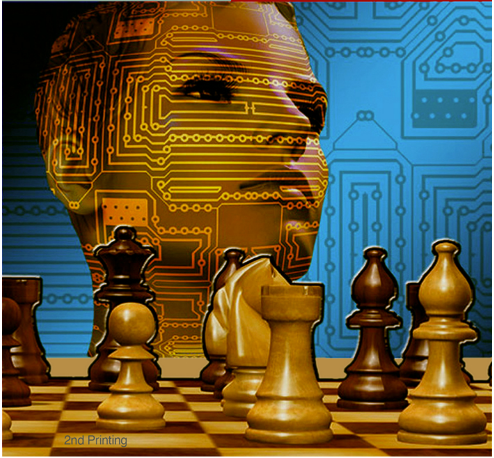


Image: CNET, 2016

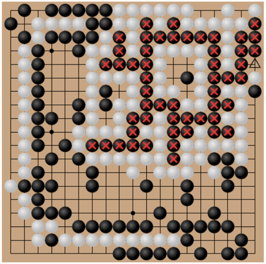
Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control

Dimitri P. Bertsekas



Adversarial Policy Beat Superhuman GO AIs

"...our adversaries do not win by learning to play Go better than KataGo. In fact, our adversaries are easily beaten by human amateurs. Instead, our adversaries win by tricking KataGo into making serious blunders..."

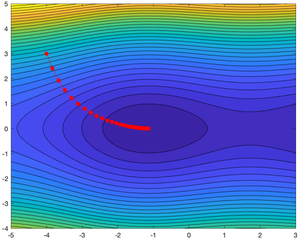


(a) Our *cyclic-adversary* wins as white by capturing a cyclic group (X) that the victim (Latest$_{\text{def}}$, 10 million visits) leaves vulnerable. Explore the game.

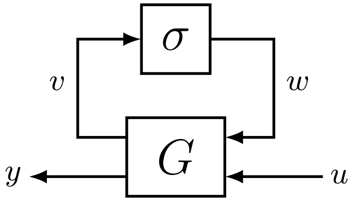
Wang et. al., arXiv:2211.00241.

Today's Goal

Static and dynamic models which are:



Compatible with ML tools (autodiff, SGD)



Compatible with nonlinear & robust stability theory (IQC)

Adversarial Inputs and Lipschitz Bounds

- **Adversarial perturbations** are small input perturbations leading to large input perturbations
- If a model $x \mapsto y$ satisfies a **Lipschitz bound**:

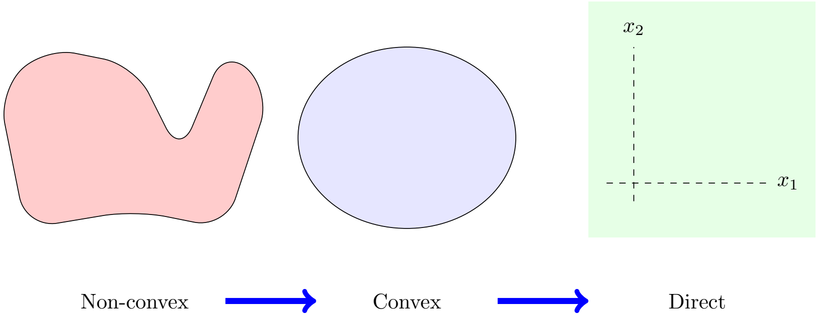
$$\|y^a - y^b\| \leq \gamma \|x^a - x^b\|$$

  then the effect of adversarial perturbations is bounded.
- The **Lipschitz constant** $\mathrm{Lip}(f)$ is defined as

$$\mathrm{Lip}(f) := \inf \left\{ L : \|f(x^a) - f(x^b)\| \leq \gamma \|x^a - x^b\|,\ \forall x^a, x^b \right\}$$

- For neural networks, exact computation of $\mathrm{Lip}(f)$ is NP-hard.

Direct Parameterizations



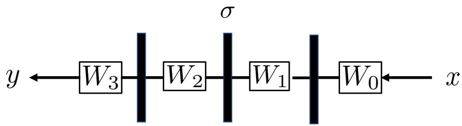
- How to impose $\mathrm{Lip}(f) \leq \gamma$ during training?
- Our approach: construct **direct** parameterization of models satisfying this bound.
  - smooth mapping from $\mathbb{R}^N$ to a set of models with $\mathrm{Lip}(f) \leq \gamma$.
  - a.k.a. an intrinsic parameterization of the constraint manifold.
- Learn via unconstrained optimization: SGD, ADAM, etc.

Robust Neural Networks

Lipschitz bound estimation for DNN

- A DNN $f : \mathbb{R}^n \to \mathbb{R}^m$ is of the form

$$f_\theta(x) = W_L \sigma(W_{L-1} \sigma(\cdots \sigma(W_0 x))) \tag{1}$$

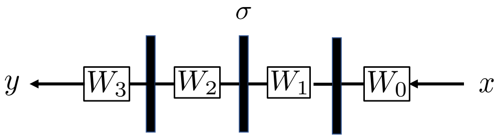


  - θ are the learnable parameters
  - σ is a fixed scalar nonlinear activations
- The **Lipschitz constant** $\mathrm{Lip}(f)$ is defined as

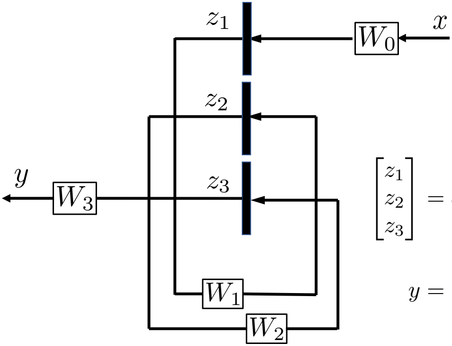
$$\mathrm{Lip}(f) := \inf \{L : \|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|,\ \forall x_1, x_2\}$$

- Computing $\mathrm{Lip}(f_\theta)$ exactly is NP hard.

Different Viewpoint on DNN Structure



"Pull apart" the weights and activations:

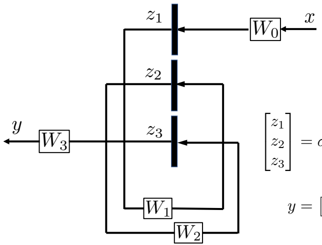


1 layer + Feedback → Deep

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \sigma\left(\begin{bmatrix} 0 & & \\ W_1 & 0 & \\ 0 & W_2 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} W_0 \\ 0 \\ 0 \end{bmatrix} x\right)$$

$$y = \begin{bmatrix} 0 & 0 & W_3 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

DNNs as Feedback Interconnections

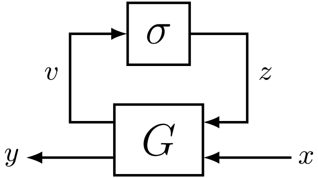


1 layer + Feedback → Deep

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \sigma\left(\begin{bmatrix} 0 & & \\ W_1 & 0 & \\ 0 & W_2 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} W_0 \\ 0 \\ 0 \end{bmatrix} x \right)$$

$$y = \begin{bmatrix} 0 & 0 & W_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Group the linear and nonlinear parts:



$$z = \sigma(v)$$

$$G: v = Wz + Ux, \quad y = Yz$$

More general structure: equilibrium (aka implicit) network

IQC Analysis Flow

- The possible input/output pairs are $(x, y) \in S$ (nasty set)
- What you want to prove about these pairs:

$$q^\star(x, y) \geq 0, \quad \forall x, y \in S$$

- What you know about S:

$$q^i(x, y) \geq 0, \quad \forall i, \quad \forall x, y \in S,$$

- Find some multipliers $\lambda_i \geq 0$, such that

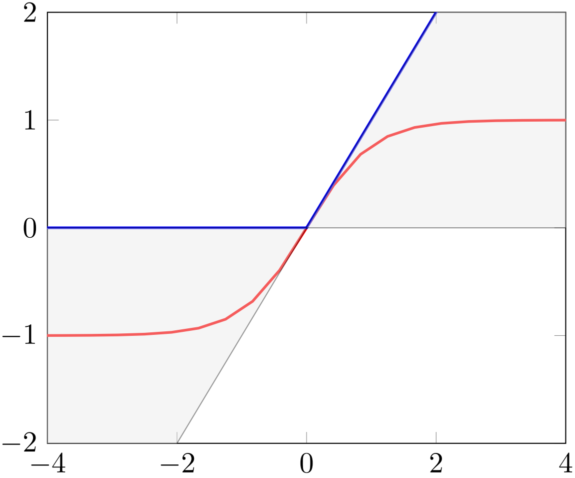
$$q^\star(x, y) \geq \sum_i \lambda_i q^i(x, y), \quad \forall (x, y)$$

  can be verified (usually SDP)
- Then

$$(x, y) \in S \implies q^i(x, y) \geq 0, \forall i \implies q^\star(x, y) \geq 0$$

Incremental Sector Bounds for $\sigma(\cdot)$

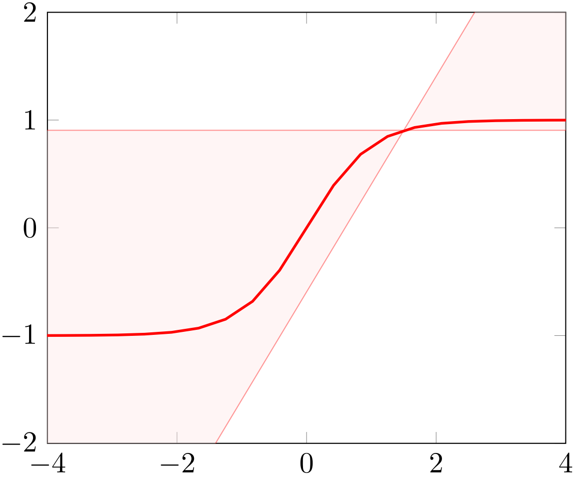
Common activations include $z = \tanh(v)$, the rectified linear unit (ReLU): $z = \max(v, 0)$



Most activations satisfy the incremental sector bound: $0 \leq \frac{\Delta z}{\Delta v} \leq 1$

Incremental Sector Bounds for $\sigma(\cdot)$

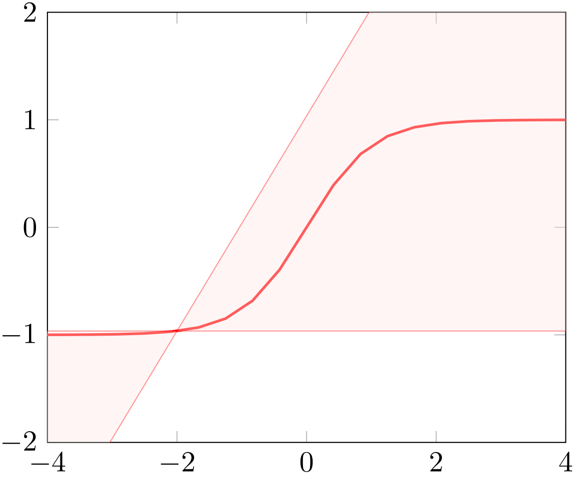
Common activations include $z = \tanh(v)$, the rectified linear unit (ReLU): $z = \max(v, 0)$



Most activations satisfy the incremental sector bound: $0 \leq \frac{\Delta z}{\Delta v} \leq 1$

Incremental Sector Bounds for $\sigma(\cdot)$

Common activations include $z = \tanh(v)$, the rectified linear unit (ReLU): $z = \max(v, 0)$



Most activations satisfy the incremental sector bound: $0 \leq \frac{\Delta z}{\Delta v} \leq 1$

Incremental Sector Bounds

- Each activation function i satisfies :

$$0 \le \frac{\Delta_z^i}{\Delta_v^i} \le 1, \quad \forall\, \Delta_v^i \ne 0$$

- This can be rewritten as

$$\Delta_z^i(\Delta_v^i - \Delta_z^i) \ge 0$$

- Since this holds for *all* activation functions, we can also take positive combinations of these, with $\lambda_i > 0$:

$$\sum_i \lambda_i \Delta_z^i(\Delta_v^i - \Delta_z^i) \ge 0$$

- Collecting terms with $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_p)$

$$2\Delta_z^T \Lambda(\Delta_v^i - \Delta_z^i) \ge 0.$$

Apply the S-Procedure

- Network representation:

$$z = \sigma(v), v = Wz + Ux, \quad y = Yz$$

- Lipschitz bound via multipliers (S-Procedure)

$$\underbrace{\gamma|\Delta_x|^2 - \frac{1}{\gamma}|\Delta_y|^2}_{\text{Lipschitz condition}} \geq \underbrace{2\Delta_z^T\Lambda(\overbrace{W\Delta_z + U\Delta_x}^{\Delta_v} - \Delta_z)}_{\geq 0 \text{ due to sector bounds}} \qquad (2)$$

- Take Schur complement and write as SDP[1] :

$$H = \begin{bmatrix} \gamma I & -U^\top\Lambda & \\ -\Lambda U & 2\Lambda - \Lambda W - W^\top\Lambda & Y^\top \\ & Y & \gamma I \end{bmatrix} \succeq 0$$

[1] Fazlyab et al. NeurIPS2019

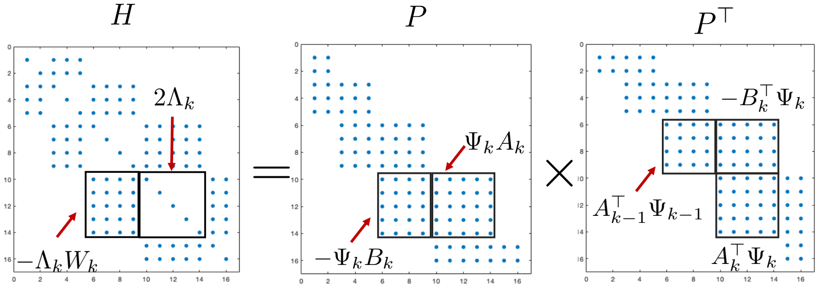
Direct Parameterization

- Basic idea: $H \succeq 0 \Leftrightarrow H = PP^\top$
- Problem: construct P s.t. H has the right sparsity structure:

$$H = \begin{bmatrix} \gamma I & -\hat{W}_0^\top & & & \\ -\hat{W}_0 & 2\Lambda_0 & -\hat{W}_1^\top & & \\ & \ddots & \ddots & \ddots & \\ & & -\hat{W}_{L-1} & 2\Lambda_{L-1} & -\hat{W}_L^\top \\ & & & -\hat{W}_L & \gamma I \end{bmatrix} \qquad (3)$$

The main diagonal blocks $\gamma I, 2\Lambda_0, 2\Lambda_1, \ldots$ are diagonal matrices.

Direct Parameterization via Cayley Transform



- Let $\Psi_k = \sqrt{\Lambda_k}$ be positive diagonal
- We need $[\, A_k \; B_k \,]$ semi-orthogonal: $A_k A_k^\top + B_k B_k^\top = I$.
- This can be parameterized directly via the Cayley transform:

$$\begin{aligned}\begin{bmatrix} A_k & B_k \end{bmatrix} &= \mathbf{cayley}(X_k, Y_k) \\ &:= \begin{bmatrix} (I - Z)(I + Z)^{-1} & -2(I + Z)^{-1} Y^\top \end{bmatrix}\end{aligned}$$

  where $Z = X_k - X_k^\top + Y_k^\top Y_k$
- X_k, Y_k are **free variables.**

Complete Parameterization

This construction provides a *complete* direct parameterization of all networks satisfying the SDP condition of Fazlyab et al (2019):

Theorem

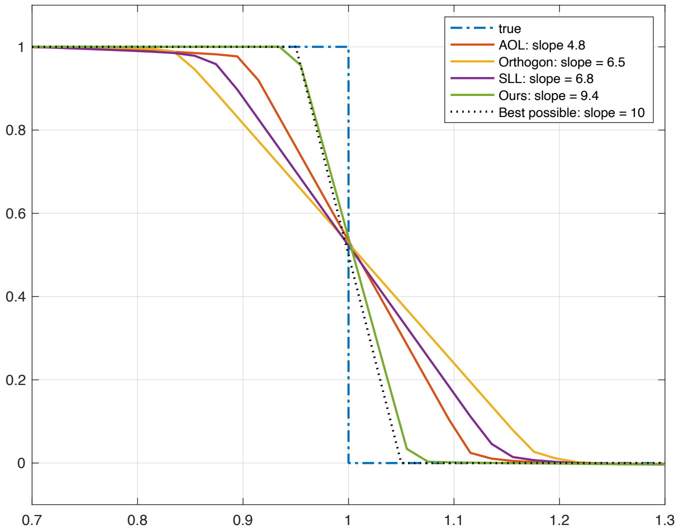
The following two conditions are eqivalent:

1. *A network with weights W satisfies the SDP $H \succeq 0$ from Fazlyab et al (2019).*
2. *The weights can be constructed as*

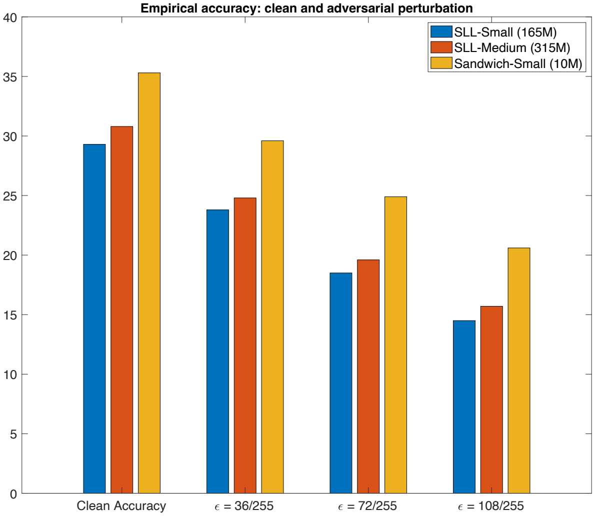
$$W_k = 2\Psi_k^{-1} B_k A_{k-1}^\top \Psi_{k-1}$$

with A, B constructed via the Cayley transform and Ψ_k positive diagonal.

Tightness: fitting a squarewave, imposing slope ≤ 10



Empirical robustness comparison on Tiny-Imagenet



Wang & Manchester, ICML 2023. SLL method: Araujo et al, ICLR 2023.

Contracting and Lipschitz Dynamic Models

What is contraction?

- A contracting dynamical system

$$x_{t+1} = f(x_t, t)$$

Has the property that *all solutions* converge exponentially

- I.e. there exists $K, \lambda > 0$ such that:

$$|x_t^a - x_t^b| \leq Ke^{-\lambda t}|x_0^a - x_0^b|.$$

- Under mild assumptions, equivalent to:

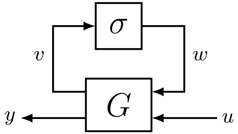
$$\sum_{t=0}^{\infty} |x_t^a - x_t^b|^2 \leq d(x_0^a, x_0^b)$$

where d is a distance metric.

- Can be interpreted as a Lipschitz condition on the mapping $x_0 \mapsto \{x_1, x_2, x_3, ...\}$

Recurrent Equilibrium Networks (REN)

A REN is an interconnection of a linear system G and nonlinear elementwise "activation functions" σ:



$$\left.\begin{aligned} x_+ &= Ax + B_1 w + B_2 u + b_x \\ v &= C_1 x + D_{11} w + D_{12} u + b_v \\ y &= C_2 x + D_{21} w + D_{22} u + b_y \end{aligned}\right\} = G, \quad w = \sigma(v)$$

Note the nonlinear *equilibrium* (a.k.a. implicit) network:

$$v_t = C_1 x + D_{11}\sigma(v_t) + D_{12}u_t + b_v$$

Can be interpreted as singular perturbation (slow/fast) model.

Model Parametrization - Model expressiveness

The REN contains many commonly used model structures:

$$\begin{bmatrix} x_{t+1} \\ v_t \\ y_t \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} x_t \\ \sigma(v_t) \\ u_t \end{bmatrix} + \begin{bmatrix} b_x \\ b_v \\ b_y \end{bmatrix}$$

- Linear Time Invariant Systems
- Recurrent Neural Networks
- Equilibrium Networks
  - Feedforward Neural Networks, Residual Networks, solutions of convex optimization problems,...
- Block oriented models
  - Wiener-Hammerstein, Hammerstein-Wiener,...

Model Parametrization - Model expressiveness

The REN contains many commonly used model structures:

$$\begin{bmatrix} x_{t+1} \\ v_t \\ y_t \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} x_t \\ \sigma(v_t) \\ u_t \end{bmatrix} + \begin{bmatrix} b_x \\ b_v \\ b_y \end{bmatrix}$$

- Linear Time Invariant Systems
- Recurrent Neural Networks: $x_{t+1} = B_1 \sigma(C_1 x_t + D_{12} u_t + b_v)$
- Equilibrium Networks
  - Feedforward Neural Networks, Residual Networks, solutions of convex optimization problems,...
- Block oriented models
  - Wiener-Hammerstein, Hammerstein-Wiener,...

Model Parametrization - Model expressiveness

The REN contains many commonly used model structures:

$$\begin{bmatrix} x_{t+1} \\ v_t \\ y_t \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} x_t \\ \sigma(v_t) \\ u_t \end{bmatrix} + \begin{bmatrix} b_x \\ b_v \\ b_y \end{bmatrix}$$

- Linear Time Invariant Systems
- Recurrent Neural Networks
- Equilibrium Networks: $v_t = D_{11}\sigma(v_t) + D_{12}u_t + b_v$
  - Feedforward Neural Networks, Residual Networks, solutions of convex optimization problems...
- Block oriented models
  - Wiener-Hammerstein, Hammerstein-Wiener,...

Model Parametrization - Model expressiveness

The REN contains many commonly used model structures:

$$\begin{bmatrix} x_{t+1} \\ v_t \\ y_t \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} x_t \\ \sigma(v_t) \\ u_t \end{bmatrix} + \begin{bmatrix} b_x \\ b_v \\ b_y \end{bmatrix}$$

- Linear Time Invariant Systems
- Recurrent Neural Networks
- Equilibrium Networks
  - Feedforward Neural Networks, Residual Networks, solutions of convex optimization problems,...
- Block oriented models
  - Wiener-Hammerstein, Hammerstein-Wiener,...

Direct Parameterization for Contraction

- Convex contraction condition

$$\underbrace{\Delta_+^T P \Delta_+ - \Delta^T P \Delta}_{\text{Metric decrease}} - \underbrace{2\Delta_+^T (E\Delta_+ - F\Delta - \tilde{B}\Delta_w)}_{=0 \text{ due to linear block}}$$
$$+ \underbrace{2\Delta_w^T (\tilde{C}\Delta + \tilde{D}_{11}\Delta_w - \Lambda\Delta_w)}_{\geq 0 \text{ due to sector condition}} \leq -\epsilon|\Delta|^2$$

- This can be written as an LMI in terms of model parameters:

$$H = \begin{bmatrix} (E + E^T - P) & -F & -\tilde{B} \\ -F^T & P & -\tilde{C}^T \\ -\tilde{B}^T & -\tilde{C} & (2\Lambda - \tilde{D}_{11} - \tilde{D}_{11}^T) \end{bmatrix} \succ 0.$$

- Lower-right term: well-posedness of the equilibrium network
  - Can be interpreted as contraction of "fast" dynamics
- **Direct parameterization:** via $H = PP^T + \epsilon I$

Lipschitz Bounds and Dissipation

- The same idea can be used to guarantee model **robustness**

$$\sum_{t=0}^{\infty} |y_t^a - y_t^b|^2 \leq \gamma \sum_{t=0}^{\infty} |u_t^a - u_t^b|^2$$

  Here γ is a *Lipschitz bound* (a.k.a. incremental ℓ^2 gain)
- Verified via the dissipation inequality

$$\Delta_+^T P \Delta_+ - \Delta^T P \Delta \leq \gamma |\Delta_u|^2 - |\Delta_y|^2$$

  where $\Delta_u = u_t^a - u_t^b$ and $\Delta_y = y_t^a - y_t^b$.
- More generally:*incremental dissipativity*

$$\sum_t \begin{bmatrix} \Delta_u \\ \Delta_y \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} \Delta_u \\ \Delta_y \end{bmatrix} \geq 0$$

- Includes incremental gain (aka Lipschitz), incremental passivity (aka monotonicity), more general IQC

Applications

Nonlinear Observer Design

- Given a nonlinear system of the form

$$x_{t+1} = f_m(x_t, u_t), \quad y_t = g_m(x_t, u_t)$$

  A standard structure is an observer of the form

$$\hat{x}_{t+1} = f_m(\hat{x}_t, u_t) + l(\hat{x}_t, u_t, y_t)$$

- Includes EKF and many other designs as special cases.
- How to design l for global stability and good statistical properties performance?
- **Generally difficult and problem dependent**

Contracting Observers: a New Paradigm

Theorem

Given a nonlinear system:

$$x_{t+1} = f_m(x_t, u_t), \quad y_t = g_m(x_t, u_t)$$

Construct an observer of the form

$$\hat{x}_{t+1} = f_o(\hat{x}_t, u_t, y_t) \tag{4}$$

such that:

1. *The system* (4) *is contracting*
2. *The following "correctness" condition holds for all* x, u*:*

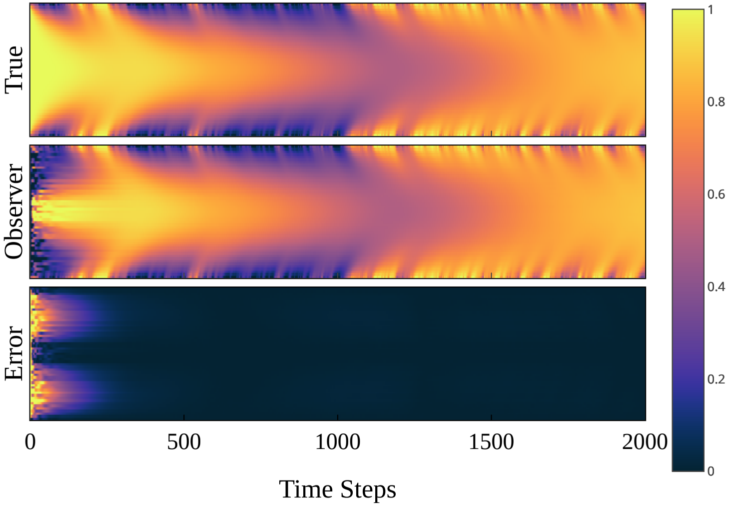
$$f_m(x, u) = f_o(x, u, g_m(x, u))$$

i.e. solutions of the true system are solutions of the observer

Then $\hat{x}_t \to x_t$ *as* $t \to \infty$.

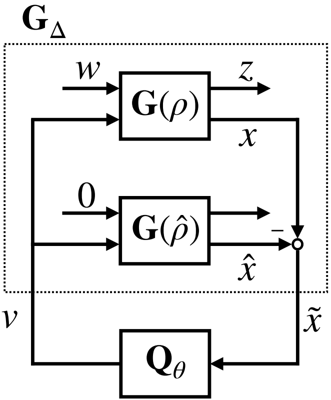
Reaction Diffusion PDE

Nonlinear Unstable PDE: $\partial_t \xi = \partial_{zz}\xi + \frac{1}{2}\xi(1-\xi)(\xi - \frac{1}{2})$

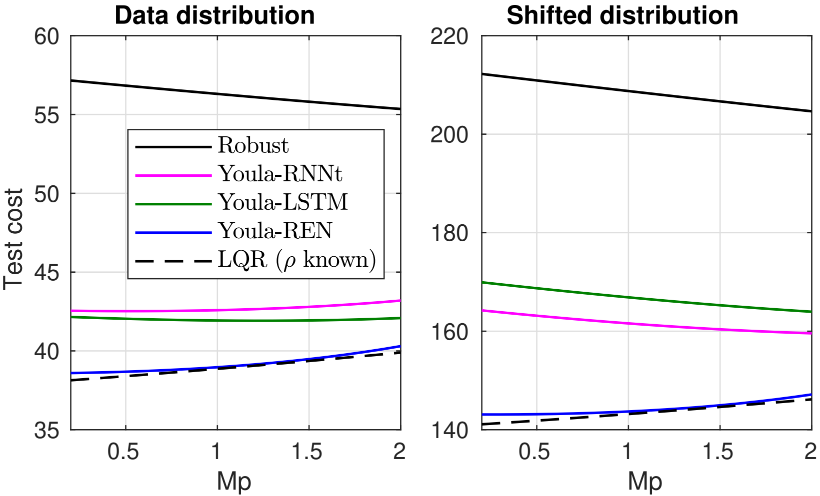


Youla-REN: Direct Adaptive Control?

- Uncertain linear model parameterized by ρ in bounded range.
- $\|G(\rho) - G(\hat{\rho})\|_\infty < \alpha$
- Q_θ: contracting nonlinear REN with Lipschitz bound $1/\alpha$.
- Train with randomized ρ



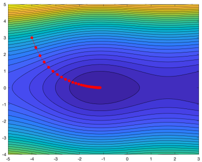
Youla-REN: Direct Adaptive Control?



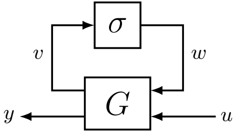
- Youla REN "adapts": without knowing ρ, performs almost as well as LQR with knowledge of ρ

Summary

We provide direct parameterizations of robust static (DNN, CNN) and dynamic (REN) models.



Direct parameterization:easily implementable with ML tools (pytorch, etc)



Incremental IQC compatible with nonlinear & robust stability theory.
Applications in SysID, observers, controllers...

Plenty more to be explored...

References etc

- **Postdoc opportunity** (opening soon)
  - Manchester, Shi, Proutiere (KTH), Megretski (MIT), "Robust Data-Driven Control for Safety-Critical Systems".
- Main papers:
  - M. Revay, R. Wang, & I. Manchester, "Recurrent Equilibrium Networks: Flexible Dynamic Models with Guaranteed Stability and Robustness", IEEE TAC (accepted), arXiv:2104.05942
  - R. Wang, N. Barbara, M. Revay, & I. Manchester, "Learning over All Stabilizing Nonlinear Controllers for a Partially-Observed Linear System", IEEE CSS Letters, 2022.
  - R. Wang & I. Manchester, "Direct Parameterization of Lipschitz Bounded Deep Networks", ICML 2023 (Oral). arXiv:2301.11526
- Tutorial paper:
  - "Contraction - Based Methods for Stable Identification and Robust Machine Learning: a Tutorial", CDC21, arXiv:2110.00207
- Related upcoming presentations:
  - JuliaCon 2023 (MIT): new Julia package: RobustNeuralNetworks.jl