International Graduate School on Control, Stuttgart, May 2024

Dissipation Inequalities and Quadratic Constraints for Control, Optimization, and Learning

Lesson 7: Applications to Neural Networks and Differential-Algebraic Equations

Murat Arcak[1] and Peter Seiler[2]

[1] University of California, Berkeley

[2] University of Michigan, Ann Arbor

Learning Objectives

In this lesson you will learn to

- Represent a neural network as an LFT with the activation functions separate from the weights and biases.
- Define quadratic constraints for common activation functions.
- Use dissipation inequalities and quadratic constraints to analyze the stability and performance of feedback systems with neural network controllers.
- Design neural network controllers
- Use dissipation inequalities and quadratic constraints to analyze the stability and performance of differential-algebraic equations

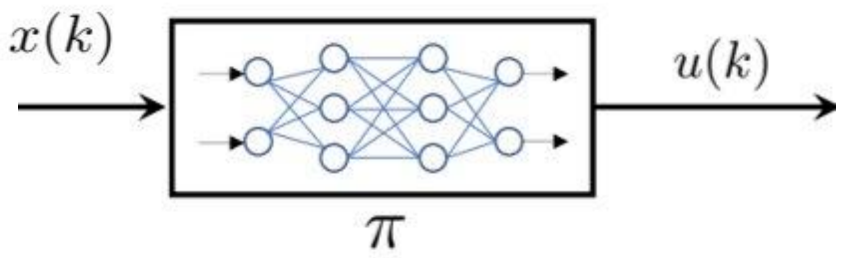
Outline

1. LFT Representations of Neural Networks
2. Quadratic Constraints for Activation Functions
3. Analysis of Neural Network Controllers
4. Synthesis of Neural Network Controllers
5. Differential-Algebraic Equations (DAEs)

LFT Representations of Neural Networks

Feedforward Neural Network

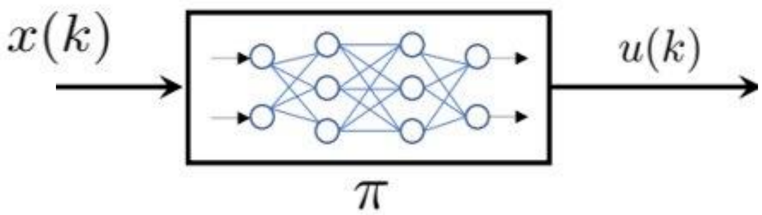
Input $x(k)$, output $u(k)$, ℓ layers.

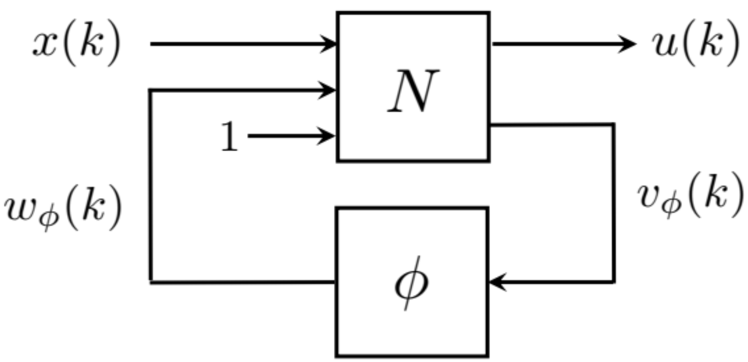


$$
\begin{aligned}
w^0(k) &= x(k), \\
w^i(k) &= \phi^i\left(W^i w^{i-1}(k) + b^i \right), \; i = 1, \ldots, \ell, \\
u(k) &= W^{\ell+1} w^\ell(k) + b^{\ell+1},
\end{aligned}
$$

Feedforward Neural Network

- Isolate the nonlinear activation functions





$$N := \left[\begin{array}{c|ccccc|c} 0 & 0 & 0 & \cdots & W^{\ell+1} & & b^{\ell+1} \\ \hline W^1 & 0 & \cdots & 0 & 0 & & b^1 \\ 0 & W^2 & \cdots & 0 & 0 & & b^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & W^\ell & 0 & & b^\ell \end{array}\right]$$

Implicit Neural Network (INN)

A typical INN formulation:

$$\hat{y}(u) = Cx + Du$$

$$x = \phi(Ax + Bu)$$

- x is defined as the fixed-point of the above equation.
- (A, B, C, D) are the trainable parameters.

Reference: El Ghaoui, et al., Implicit Deep Learning, SIAM, 2021.

Implicit Neural Network

Modeling a dense feedforward NN with L layers:

$$\hat{y} = W_L x_L + b_L, \quad x_{l+1} = \phi_l(W_l x_l + b_l), \quad x_0 = u$$

First define $x = (x_1, \ldots, x_L)$ and $\phi = (\phi_0, \ldots, \phi_{L-1})$

Then,

$$\hat{y}(u) = \underbrace{[0 \ \ldots \ 0 \ W_L]}_{C} x + \underbrace{[0 \ b_L]}_{D} \begin{bmatrix} u \\ 1 \end{bmatrix}$$

$$x = \phi\left(\underbrace{\begin{bmatrix} 0 & & & & \\ W_1 & 0 & & & \\ 0 & W_2 & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \ldots & 0 & W_{L-1} & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} W_0 & b_0 \\ 0 & b_1 \\ \vdots & \vdots \\ 0 & b_{L-1} \end{bmatrix}}_{B} \begin{bmatrix} u \\ 1 \end{bmatrix} \right)$$

Well-Posedness of INNs

$$x = \phi(Ax + Bu)$$

When does a fixed point exist, and when is it unique?

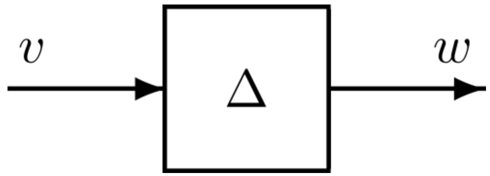
- Depends on structure of A; many conditions possible.

- A useful condition for our method [1]:
  - Search for diagonal $\Lambda \succ 0$ such that $\Lambda A + A^\top \Lambda - 2\Lambda \prec 0$.

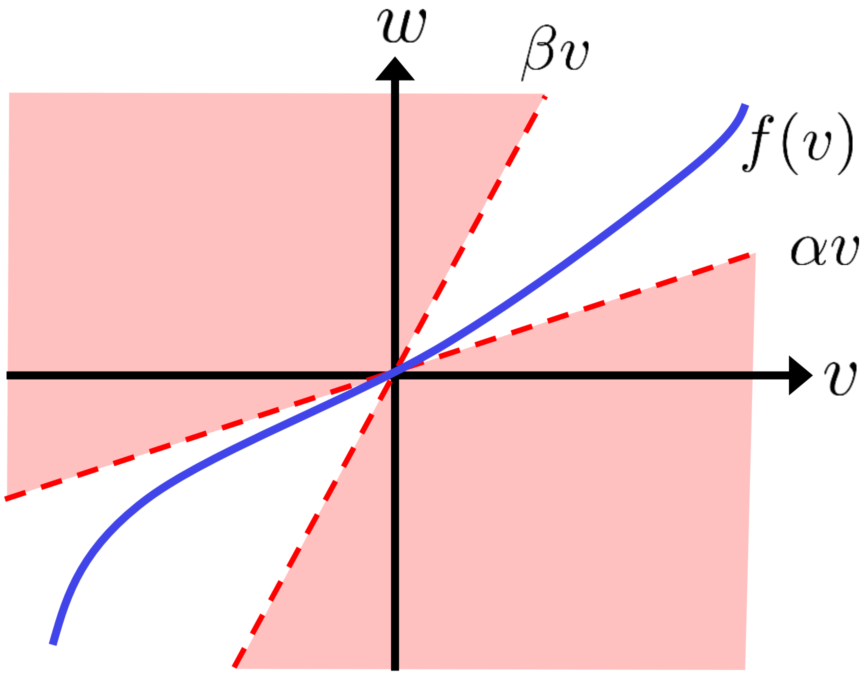
[1] Revay, Wang, Manchester, Recurrent Equilibrium Networks: Flexible Dynamic Models With Guaranteed Stability and Robustness, TAC, 2024.

Quadratic Constraints for Activation Functions

Example: Sector-bounded Nonlinearity



Suppose Δ is a nonlinearity, $w = f(v)$, whose graph lies in the sector $[\alpha, \beta]$.



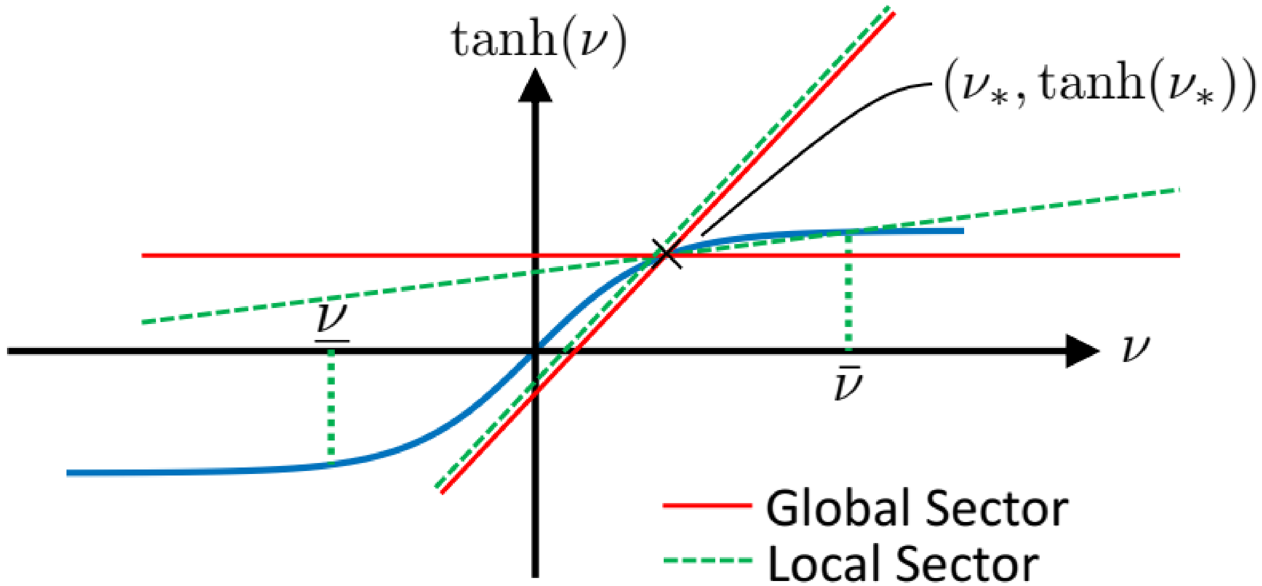
$$(w(t) - \alpha v(t)) \cdot (\beta v(t) - w(t)) \geq 0$$

$$\begin{bmatrix} v(t) \\ w(t) \end{bmatrix}^\top \underbrace{\begin{bmatrix} -2\alpha\beta & \alpha + \beta \\ \alpha + \beta & -2 \end{bmatrix}}_{:= J} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \geq 0$$

Δ satisfies the static QC defined by J.

Sector Bounds

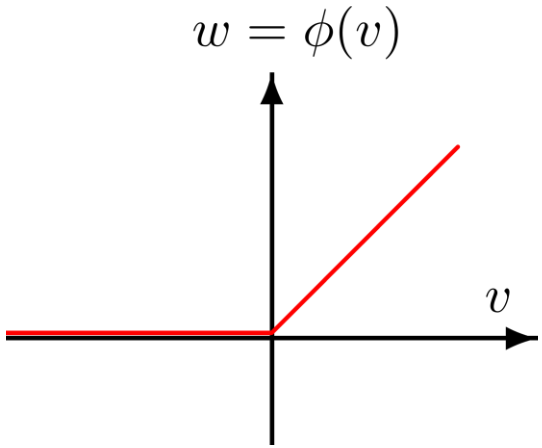
Local quadratic constraints on the activation function.



Scalar Rectified Linear Unit (ReLU)

Scalar ReLU is $\phi: \mathbb{R} \to \mathbb{R}_{\geq 0}$ is:

$$\phi(v) = \begin{cases} 0 & \text{if } v < 0 \\ v & \text{if } v \geq 0 \end{cases}$$

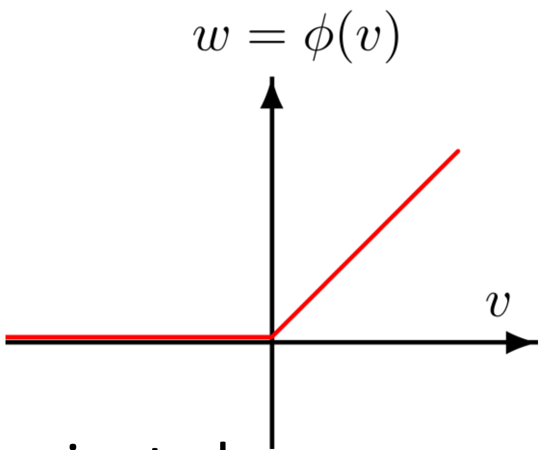


ϕ is sector and slope constrained to [0,1].

Scalar Rectified Linear Unit (ReLU)

Scalar ReLU is $\phi: \mathbb{R} \to \mathbb{R}_{\geq 0}$ is:

$$\phi(v) = \begin{cases} 0 & \text{if } v < 0 \\ v & \text{if } v \geq 0 \end{cases}$$



ϕ is sector and slope constrained to [0,1].

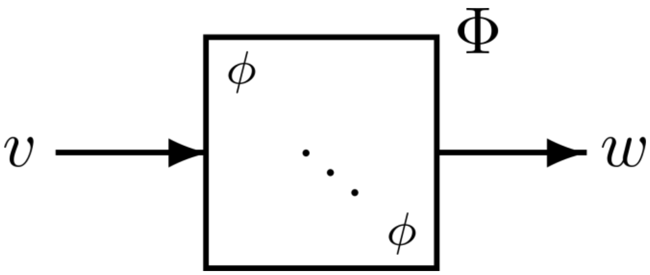
In addition, it satisfies (Richardson, et al.; Ebhihari, et al.; Drummond, et al.; Fazlyab, et al.):

- *Positivity:* $\phi(v) \geq 0 \quad \forall v \in \mathbb{R}$.
- *Positive Complement:* $\phi(v) \geq v \quad \forall v \in \mathbb{R}$.
- *Complementarity:* $\phi(v)\big(v - \phi(v)\big) = 0 \quad \forall v \in \mathbb{R}$.
- *Positive Homogeneity:* $\phi(\beta v) = \beta\phi(v) \ \forall v \in \mathbb{R}$ and $\forall \beta \geq 0$

The properties can be used to write QCs that are specific to ReLU (in addition to sector and slope constraints).

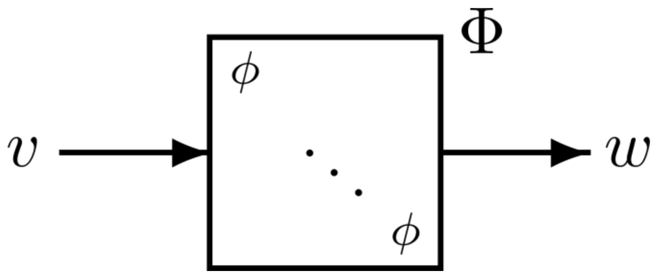
Repeated ReLU

The repeated ReLU $\Phi: \mathbb{R}^m \to \mathbb{R}^m$ maps elementwise: $w_i = \phi(v_i)$ for $i = 1, \ldots, m$ where ϕ is the scalar ReLU.



Repeated ReLU

The repeated ReLU $\Phi: \mathbb{R}^m \to \mathbb{R}^m$ maps elementwise: $w_i = \phi(v_i)$ for $i = 1, \ldots, m$ where ϕ is the scalar ReLU.



Def: $M \in \mathbb{R}^{m\times m}$ is doubly hyperdominant if the off-diagonal elements are non-positive and the row / column sums are non-negative.

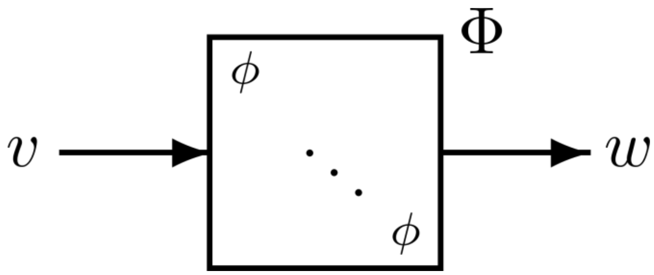
QC 1: If $Q_0 \in \mathbb{R}^{m\times m}$ is doubly hyperdominant then

$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} 0 & Q_0^\top \\ Q_0 & -(Q_0 + Q_0^\top) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \geq 0 \ \forall v \in \mathbb{R}^m \text{ and } w = \Phi(v)$$

This QC holds for any repeated function that is slope-restricted in [0,1] and passes through the origin [Willems, Brocket, ’68; Willems ’71].

Repeated ReLU

The repeated ReLU $\Phi: \mathbb{R}^m \to \mathbb{R}^m$ maps elementwise: $w_i = \phi(v_i)$ for $i = 1, \dots, m$ where ϕ is the scalar ReLU.



QC 2: If $Q_1 \in \mathbb{R}^{m\times m}$ is diagonal then

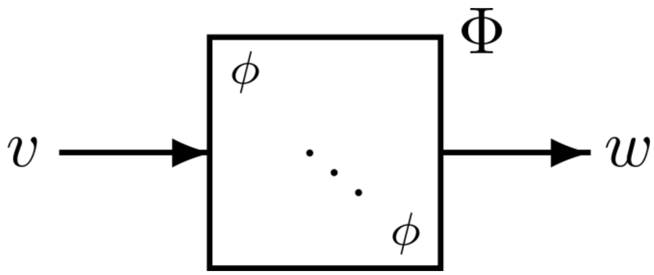
$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} 0 & Q_1 \\ Q_1 & -2Q_1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = 0 \ \forall v \in \mathbb{R}^m \text{ and } w = \Phi(v)$$

This follows from complementarity of scalar ReLU:

$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} 0 & Q_1 \\ Q_1 & -2Q_1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \sum_{k=1}^{m} (Q_1)_{kk} w_k (v_k - w_k) = 0$$

Repeated ReLU

The repeated ReLU $\Phi: \mathbb{R}^m \to \mathbb{R}^m$ maps elementwise: $w_i = \phi(v_i)$ for $i = 1, \dots, m$ where ϕ is the scalar ReLU.



QC 3: If $Q_2, Q_3, Q_4 \in \mathbb{R}_{\geq 0}^{m \times m}$ with $Q_2 = Q_2^\top$ and $Q_3 = Q_3^\top$ then

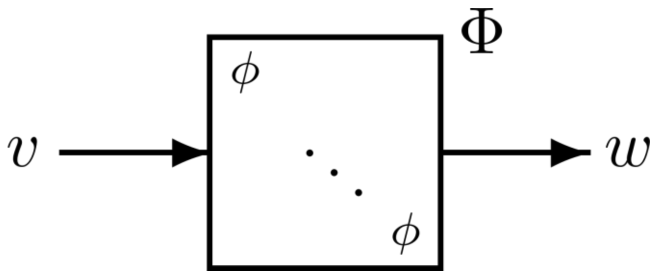
$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} Q_2 & -(Q_2 + Q_4^\top) \\ -(Q_2 + Q_4) & Q_2 + Q_3 + Q_4 + Q_4^\top \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \geq 0 \;\; \forall v \in \mathbb{R}^m \text{ and } w = \Phi(v)$$

This follows by taking combinations of the linear constraints implied by the positivity and positive complement properties:

$$(Q_2)_{kj}\,(w_k - v_k)(w_j - v_j) \geq 0,$$
$$(Q_3)_{kj}\,w_k w_j \geq 0,$$
$$(Q_4)_{kj}\,w_k(w_j - v_j) \geq 0$$

Repeated ReLU

The repeated ReLU $\Phi: \mathbb{R}^m \to \mathbb{R}^m$ maps elementwise: $w_i = \phi(v_i)$ for $i = 1, \dots, m$ where ϕ is the scalar ReLU.



Def: $M \in \mathbb{R}^{m \times m}$ is Metzler matrix if the off-diag. elements are ≥ 0.

QC: If $Q_2 = Q_2^\top, Q_3 = Q_3^\top \in \mathbb{R}_{\geq 0}^{m \times m}$ & $\tilde{Q} \in \mathbb{R}^{m \times m}$ is Metzler matrix then

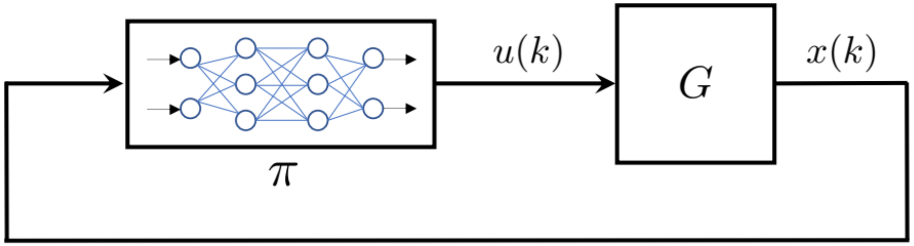
$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} Q_2 & -\tilde{Q}^\top - Q_2 \\ -\tilde{Q} - Q_2 & Q_2 + Q_3 + \tilde{Q} + \tilde{Q}^\top \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \geq 0 \ \ \forall v \in \mathbb{R}^m \text{ and } w = \Phi(v)$$

This is the largest class of QCs for the known properties of scalar ReLU. Positive homogeneity does not increase the class of QCs (Vahedi-Noori, et al, arXiv, '24).

Analysis of Neural Network Controllers

ROA Problem Formulation

- Plant *G* is LTI & Neural Network π is a static, state-feedback.



- Neural-network has ℓ-layers:

$$
\begin{aligned}
w^0(k) &= x(k), \\
w^i(k) &= \phi^i\left(W^i w^{i-1}(k) + b^i \right), \; i = 1, \ldots, \ell, \\
u(k) &= W^{\ell+1} w^\ell(k) + b^{\ell+1},
\end{aligned}
$$

where W^i, b^i, *and* ϕ^i are the weights, biases, & activation functions.

Goal: Compute an estimate of the region of attraction (ROA) of initial conditions that converge back to the equilibrium point.

Approach:

1. Isolate the nonlinear activation functions

2. Express local quadratic constraints on the activation function.

3. Use Lyapunov theory, local quadratic constraints, and convex optimization to estimate the region of attraction.
    - Lyapunov condition also proves local region assumption used to derive quadratic constraints is valid.

Comments:

- The framework can be extended to handle nonlinearities and uncertainties in the plant G.
- This extension can be used to compute disk margins for neural network-based controllers.

Region of Attraction Condition

This is discrete time, but is analogous to continuous time.

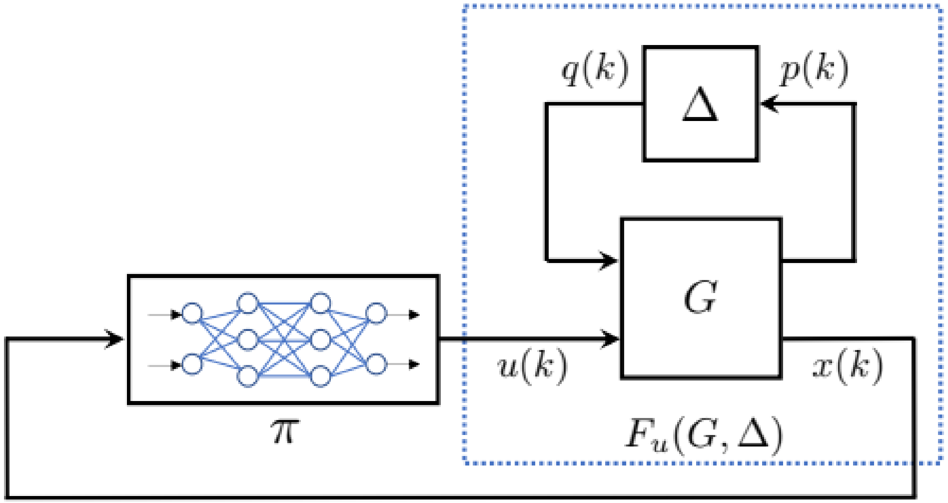
$$R_V^\top \begin{bmatrix} A_G^\top P A_G - P & A_G^\top P B_G \\ B_G^\top P A_G & B_G^\top P B_G \end{bmatrix} R_V + R_\phi^\top \Psi_\phi^\top \, J_\phi(\lambda) \, \Psi_\phi R_\phi < 0,$$

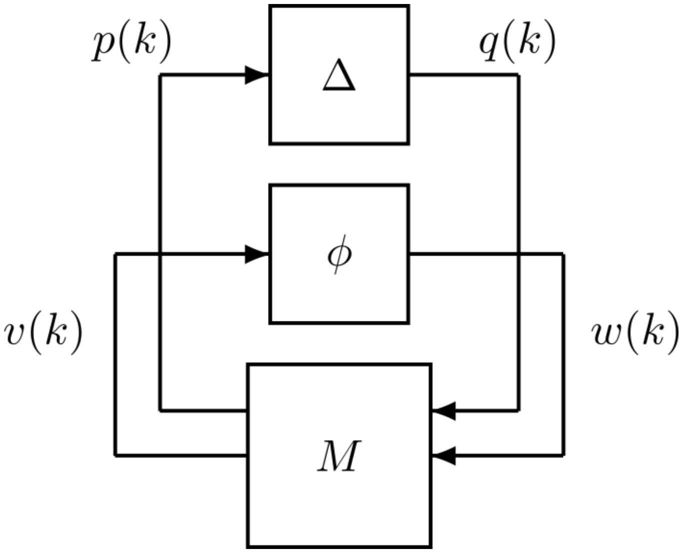
$$\begin{bmatrix} (\bar{v}_i^1 - v_{*,i}^1)^2 & W_i^1 \\ W_i^{1\top} & P \end{bmatrix} \geq 0, \; i = 1, \cdots, n_1,$$

- (A_G, B_G, C_G, D_G) are system matrices.
- $(\Psi_\phi, J_\phi(\lambda))$ are for NN activation function IQC.
- W terms are related to NN weights.

Robust Region of Attraction

We can also estimate the region of attraction when the plant is uncertain and the controller is a neural network.





Robust Region of Attration Condition

This is discrete time, but is analogous to continuous time.

$$R_V^\top \begin{bmatrix} \mathcal{A}^\top P \mathcal{A} - P & \mathcal{A}^\top P \mathcal{B} \\ \mathcal{B}^\top P \mathcal{A} & \mathcal{B}^\top P \mathcal{B} \end{bmatrix} R_V + R_\phi^\top \Psi_\phi^\top J_\phi(\lambda) \; \Psi_\phi R_\phi$$

$$+ R_V^\top \begin{bmatrix} \mathcal{C} & \mathcal{D} \end{bmatrix}^\top J_\Delta \begin{bmatrix} \mathcal{C} & \mathcal{D} \end{bmatrix} R_V < 0$$

$$\begin{bmatrix} (\bar{v}_i^1)^2 & \mathcal{W}_i^1 \\ \mathcal{W}_i^{1\top} & P \end{bmatrix} \geq 0, \; i = 1, \ldots, n_1$$

- $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ are system matrices.
- $(\Psi_\phi, J_\phi(\lambda))$ are for NN activation function IQC.
- J_Δ is for plant uncertainty IQC.
- W terms are related to NN weights.

ROA Experiments: Inverted Pendulum

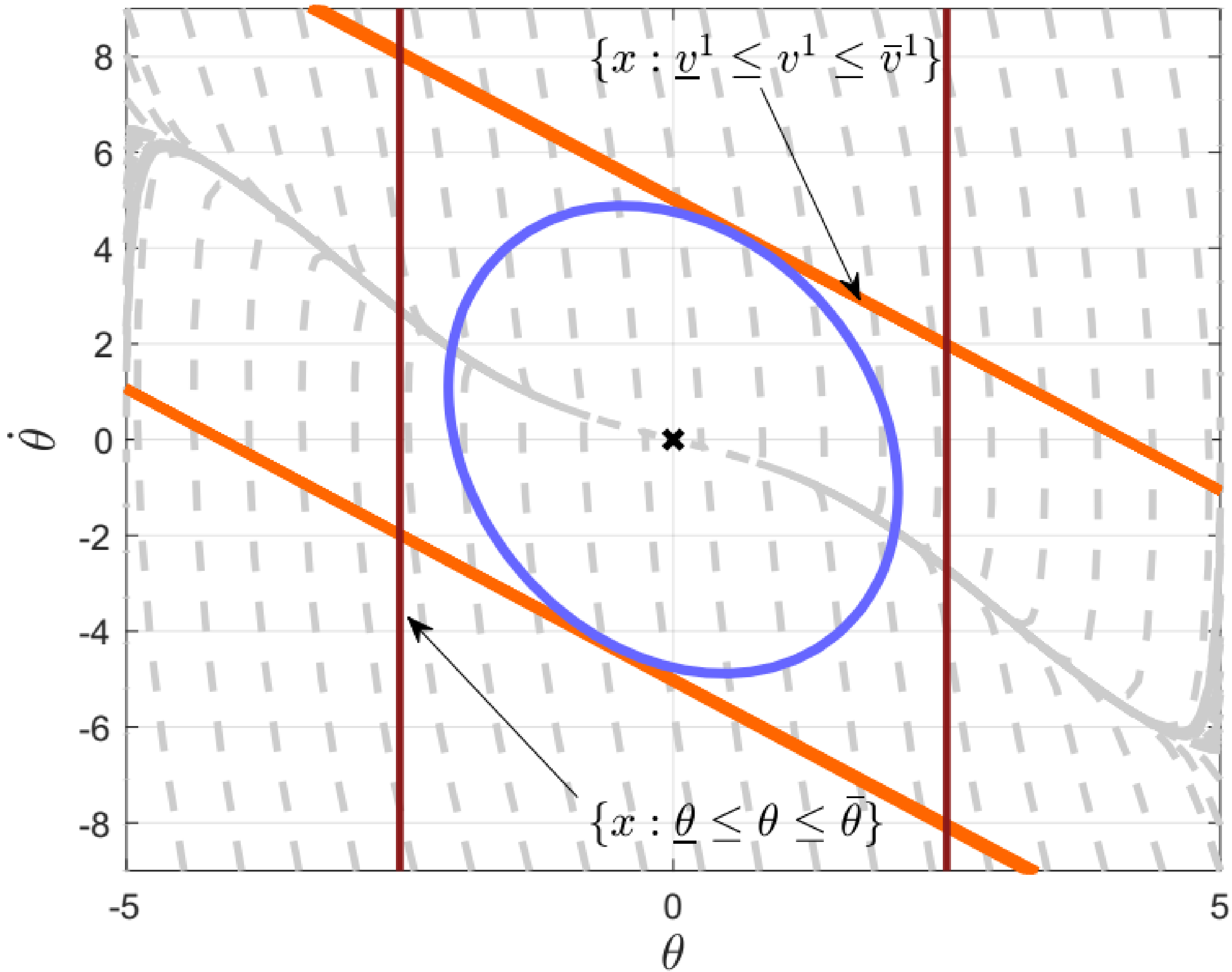
- Equations of Motion with angle θ (rad):

$$\ddot{\theta}(t) = \frac{mgl\sin(\theta(t)) - \mu\dot{\theta}(t) + u(t)}{ml^2},$$

  - mass m=0.15kg, length l = 0.5m, friction μ=0.5 Nms/rad.
  - Dynamics discretized with dt=0.02s.
  - Trigonometric terms also bounded with sector constraints

- Neural network designed via reinforcement learning
  - 2 Layers
  - 32 neurons in each layer
  - tanh as the activation function
  - All biases set to zero

ROA Experiments: Inverted Pendulum



ROA Experiments: Lateral Vehicle Control

- Equations of Motion with perp. distance to lane edge *e* (m) and e_θ is the angle between the car and lane (rad):

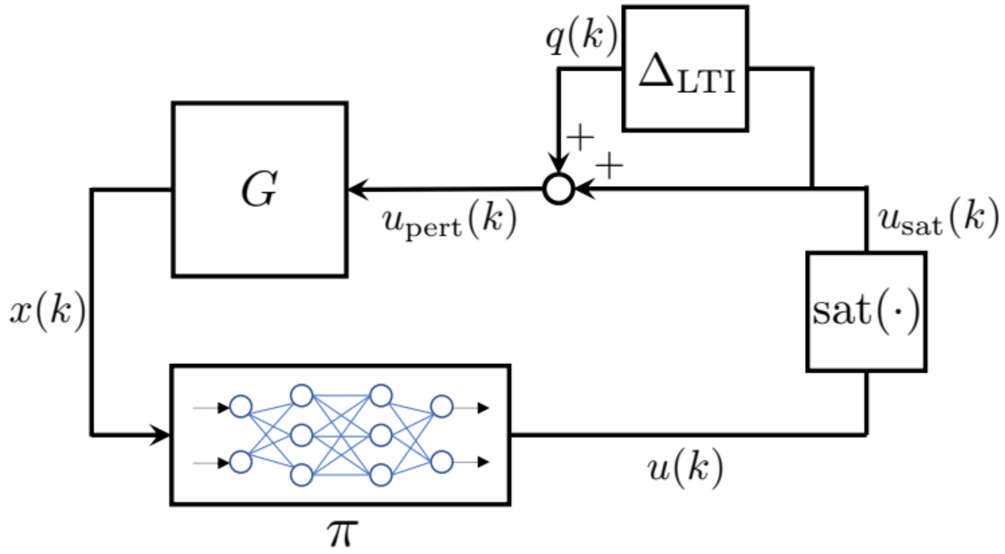
$$\begin{bmatrix} \dot{e} \\ \ddot{e} \\ \dot{e}_\theta \\ \ddot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_{\alpha f}+C_{\alpha r}}{mU} & -\frac{C_{\alpha f}+C_{\alpha r}}{m} & \frac{aC_{\alpha f}-bC_{\alpha r}}{mU} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{aC_{\alpha f}-bC_{\alpha r}}{I_z U} & -\frac{aC_{\alpha f}-bC_{\alpha r}}{I_z} & \frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{I_z U} \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ e_\theta \\ \dot{e}_\theta \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ -\frac{C_{\alpha f}}{m} \\ 0 \\ -\frac{aC_{\alpha f}}{I_z} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{aC_{\alpha f}-bC_{\alpha r}}{m} - U^2 \\ 0 \\ \frac{a^2C_{\alpha f}+b^2C_{\alpha r}}{I_z} \end{bmatrix} c \qquad (35)$$

  - Parameters given the paper.
  - Dynamics discretized with dt=0.02s.

ROA Experiments: Lateral Vehicle Control

- Equations of Motion with perp. distance to lane edge e (m) and e_θ is the angle between the car and lane (rad):
  - Parameters given the paper.
  - Dynamics discretized with dt=0.02s.
  - Saturation and unmodeled dynamics included in analysis.

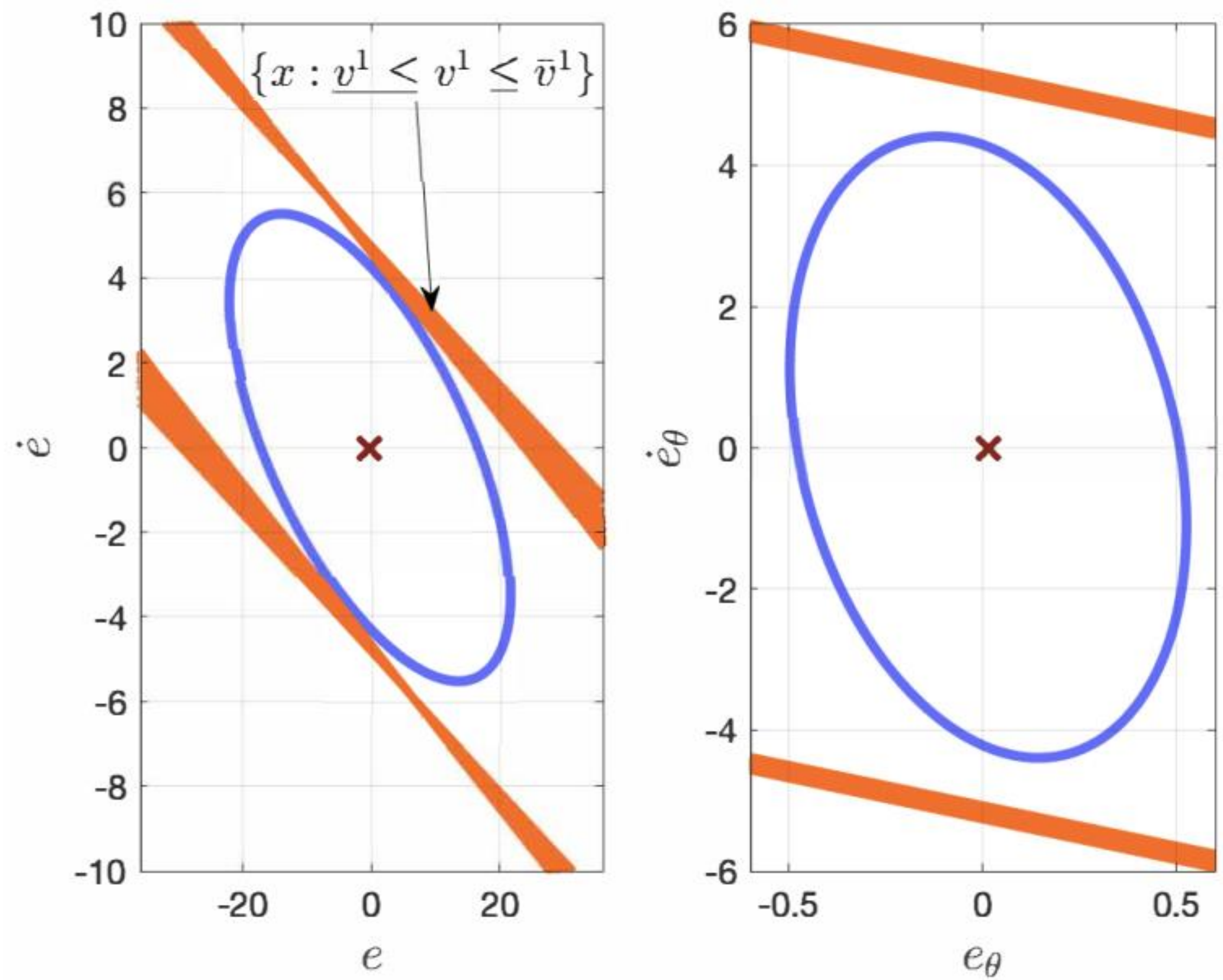


ROA Experiments: Lateral Vehicle Control

- Equations of Motion with perp. distance to lane edge e (m) and e_θ is the angle between the car and lane (rad):
  - Parameters given the paper.
  - Dynamics discretized with dt=0.02s.
  - Saturation and unmodeled dynamics included in analysis.

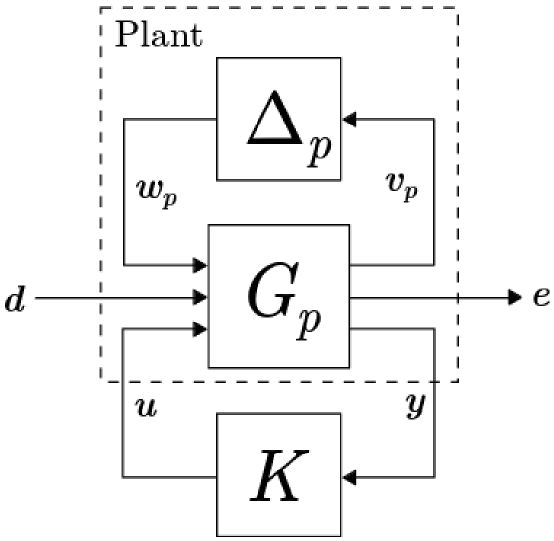
- Neural network designed via reinforcement learning
  - 2 Layers
  - 32 neurons in each layer
  - tanh as the activation function

ROA Experiments: Lateral Vehicle Control



NN Controller Performance Analysis

- Plant is interconnection of LTI system G_p and uncertainty Δ_p.
- Controller K is recurrent implicit neural network.

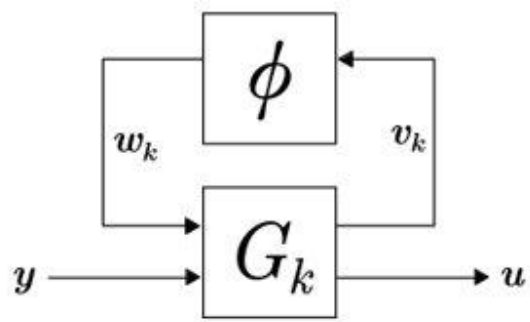


Goal: Check dissipativity (d, e).

Based on work by Junnarkar, Yin, Gu, Arcak, Seiler

Approach

1. Model plant and controller alike:
   - Interconnections of LTI systems with uncertainties



2. Characterize NN activation functions with quadratic constraints
3. Characterize plant uncertainty with IQCs
4. Construct dissipation inequality.

Plant Model

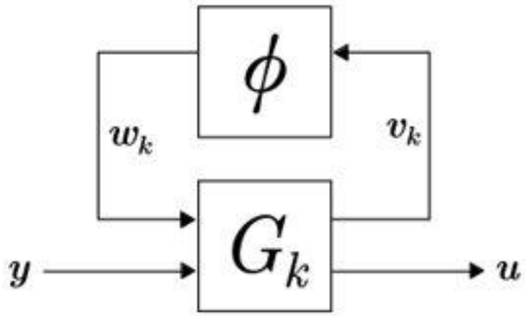
$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{p}}(t) \\ \boldsymbol{v}_{\boldsymbol{p}}(t) \\ \boldsymbol{e}(t) \\ \boldsymbol{y}(t) \end{bmatrix} = \begin{bmatrix} A_p & B_{pw} & B_{pd} & B_{pu} \\ C_{pv} & D_{pvw} & D_{pvd} & D_{pvu} \\ C_{pe} & D_{pew} & D_{ped} & D_{peu} \\ C_{py} & D_{pyw} & D_{pyd} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{p}}(t) \\ \boldsymbol{w}_{\boldsymbol{p}}(t) \\ \boldsymbol{d}(t) \\ \boldsymbol{u}(t) \end{bmatrix}$$

$$\boldsymbol{w}_{\boldsymbol{p}}(t) = \Delta_p(\boldsymbol{v}_{\boldsymbol{p}})(t),$$

Δ_p is an uncertainty, described by IQCs

Neural Network Controller Model

INN + state:



$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{k}}(t) \\ \boldsymbol{v}_{\boldsymbol{k}}(t) \\ \boldsymbol{u}(t) \end{bmatrix} = \begin{bmatrix} A_k & B_{kw} & B_{ky} \\ C_{kv} & D_{kvw} & D_{kvy} \\ C_{ku} & D_{kuw} & D_{kuy} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{k}}(t) \\ \boldsymbol{w}_{\boldsymbol{k}}(t) \\ \boldsymbol{y}(t) \end{bmatrix}$$

$$\boldsymbol{w}_{\boldsymbol{k}}(t) = \phi(\boldsymbol{v}_{\boldsymbol{k}}(t)),$$

- $\boldsymbol{w}_{\boldsymbol{k}}(t)$ is defined implicitly → implicit neural network
  - We use unbiased implicit neural networks
- "Recurrent Implicit Neural Network (RINN)"

Discrete-time Models

- Analogous conditions hold for discrete-time systems.

Plant:

$$\begin{bmatrix} \boldsymbol{x_p}[t+1] \\ \boldsymbol{v_p}[t] \\ \boldsymbol{e}[t] \\ \boldsymbol{y}[t] \end{bmatrix} = \begin{bmatrix} A_p & B_{pw} & B_{pd} & B_{pu} \\ C_{pv} & D_{pvw} & D_{pvd} & D_{pvu} \\ C_{pe} & D_{pew} & D_{ped} & D_{peu} \\ C_{py} & D_{pyw} & D_{pyd} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_p}[t] \\ \boldsymbol{w_p}[t] \\ \boldsymbol{d}[t] \\ \boldsymbol{u}[t] \end{bmatrix}$$

$$\boldsymbol{w_p}[t] = \Delta_p(\boldsymbol{v_p})[t]$$

Controller:

$$\begin{bmatrix} \boldsymbol{x_k}[t+1] \\ \boldsymbol{v_k}[t] \\ \boldsymbol{u}[t] \end{bmatrix} = \begin{bmatrix} A_k & B_{kw} & B_{ky} \\ C_{kv} & D_{kvw} & D_{kvy} \\ C_{ku} & D_{kuw} & D_{kuy} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_k}[t] \\ \boldsymbol{w_k}[t] \\ \boldsymbol{y}[t] \end{bmatrix}$$

$$\boldsymbol{w_k}[t] = \phi(\boldsymbol{v_k}[t])$$

Feedback System

- Controller model of same form as plant model:

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{k}}(t) \\ \boldsymbol{v}_{\boldsymbol{k}}(t) \\ \boldsymbol{u}(t) \end{bmatrix} = \begin{bmatrix} A_k & B_{kw} & B_{ky} \\ C_{kv} & D_{kvw} & D_{kvy} \\ C_{ku} & D_{kuw} & D_{kuy} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{k}}(t) \\ \boldsymbol{w}_{\boldsymbol{k}}(t) \\ \boldsymbol{y}(t) \end{bmatrix}$$

$$\boldsymbol{w}_{\boldsymbol{k}}(t) = \phi(\boldsymbol{v}_{\boldsymbol{k}}(t)),$$

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{p}}(t) \\ \boldsymbol{v}_{\boldsymbol{p}}(t) \\ \boldsymbol{e}(t) \\ \boldsymbol{y}(t) \end{bmatrix} = \begin{bmatrix} A_p & B_{pw} & B_{pd} & B_{pu} \\ C_{pv} & D_{pvw} & D_{pvd} & D_{pvu} \\ C_{pe} & D_{pew} & D_{ped} & D_{peu} \\ C_{py} & D_{pyw} & D_{pyd} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{p}}(t) \\ \boldsymbol{w}_{\boldsymbol{p}}(t) \\ \boldsymbol{d}(t) \\ \boldsymbol{u}(t) \end{bmatrix}$$

$$\boldsymbol{w}_{\boldsymbol{p}}(t) = \Delta_p(\boldsymbol{v}_{\boldsymbol{p}})(t),$$

- Results in feedback system of same form:

$$\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \boldsymbol{v}(t) \\ \boldsymbol{e}(t) \end{bmatrix} = \begin{bmatrix} A & B_w & B_d \\ C_v & D_{vw} & D_{vd} \\ C_e & D_{ew} & D_{ed} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{w}(t) \\ \boldsymbol{d}(t) \end{bmatrix}$$

$$\boldsymbol{w}(t) = \Delta(\boldsymbol{v})(t),$$

Dissipation Inequality

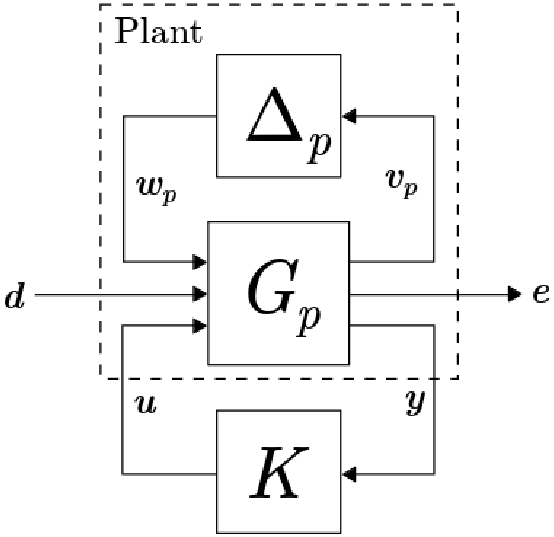
- Assume Δ satisfies a set of static IQCs: $\{J\}$.
- Assume supply rate is quadratic, parameterized by X.
- Search for $\lambda \geqslant 0$, a J, and a quadratic storage function $x^\top P x, P \succcurlyeq 0$ such that:

$$\begin{bmatrix} A^\top P + PA & PB_w & PB_d \\ B_w^\top P & 0 & 0 \\ B_d^\top P & 0 & 0 \end{bmatrix} + \lambda(\star)^\top J \begin{bmatrix} C_v & D_{vw} & D_{vd} \\ 0 & I & 0 \end{bmatrix} - (\star)^\top X \begin{bmatrix} 0 & 0 & I \\ C_e & D_{ew} & D_{ed} \end{bmatrix} \preccurlyeq 0$$

Synthesis of Neural Network Controllers

Neural Network Controller Synthesis

- Plant is interconnection of LTI system G_p and uncertainty Δ_p.
- Design controller K such that:
  - Supply rate on (d, e) is satisfied
  - Reward is maximized



$$K^* = \arg\max_K \quad \mathbb{E}\left[\int_0^T r(x(t), u(t))dt\right]$$
$$\text{s.t.} \quad K \text{ makes closed-loop dissipative}$$

Based on work by Junnarkar, Yin, Gu, Arcak, Seiler

Example Uses

- Robustness to disturbances with minimal control effort:
  - Supply rate: L_2 gain from disturbance to plant state
  - Reward: $-\|u\|^2$
- Use simulator to optimize controller with:
  - More realistic disturbances
  - Higher fidelity plant model

Approach

1. Convexify dissipation inequality.
2. Train NN controller using reinforcement learning
   - Project into certified safe set as needed.

Convexification

- Convexity important for tractable optimization.
- Previous dissipation inequality is not convex in both the controller parameters θ and the storage function P.
- Change of variables (to new variables $\hat{\theta}$) based on (Scherer, Gahinet, Chilali).

- Additional assumptions:
  - X_{ee} negative semidefinite
  - $J_{\Delta_p vv}$ positive semidefinite
- Restriction to positive definite P.

Convexification

- By Schur complement, dissipation inequality becomes:

$$\begin{bmatrix} F & \begin{bmatrix} C_v^\top L_\Delta^\top & C_e^\top L_X^\top \\ D_{vw}^\top L_\Delta^\top & D_{ew}^\top L_X^\top \\ D_{vd}^\top L_\Delta^\top & D_{ed}^\top L_X^\top \end{bmatrix} \\ \begin{bmatrix} L_\Delta C_v & L_\Delta D_{vw} & L_\Delta D_{vd} \\ L_X C_e & L_X D_{ew} & L_X D_{ed} \end{bmatrix} & -I \end{bmatrix} \preceq 0$$

$$F = \begin{bmatrix} A^\top P + PA & PB_w & PB_d \\ B_w^\top P & 0 & 0 \\ B_d^\top P & 0 & 0 \end{bmatrix} + (\star)^\top \begin{bmatrix} 0 & J_{vw} \\ J_{vw}^\top & J_{ww} \end{bmatrix} \begin{bmatrix} C_v & D_{vw} & D_{vd} \\ 0 & I & 0 \end{bmatrix} - (\star)^\top \begin{bmatrix} X_{dd} & X_{de} \\ X_{de}^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & I \\ C_e & D_{ew} & D_{ed} \end{bmatrix}$$

This is bilinear in controller parameters and storage function parameter.

Convexification

Change of variables based on (Scherer, Gahinet, Chilali).

- Introduce a partition of P and its inverse:

$$P = \begin{bmatrix} S & U \\ U^\top & \star \end{bmatrix} \quad P^{-1} = \begin{bmatrix} R & V \\ V^\top & \star \end{bmatrix}$$

$$Y \triangleq \begin{bmatrix} R & I \\ V^\top & 0 \end{bmatrix}$$

Convexification

- Left and right multiply by $Y^\top$ and Y :

$$\begin{bmatrix} \begin{bmatrix} Y^\top & \\ & I \end{bmatrix} F \begin{bmatrix} Y & \\ & I \end{bmatrix} & \begin{bmatrix} Y^\top C_v^\top L_\Delta^\top & Y^\top C_e^\top L_X^\top \\ D_{vw}^\top L_\Delta^\top & D_{ew}^\top L_X^\top \\ D_{vd}^\top L_\Delta^\top & D_{ed}^\top L_X^\top \end{bmatrix} \\ \begin{bmatrix} L_\Delta C_v Y & L_\Delta D_{vw} & L_\Delta D_{vd} \\ L_X C_e Y & L_X D_{ew} & L_X D_{ed} \end{bmatrix} & -I \end{bmatrix} \preceq 0$$

$$A^\top P + PA \longrightarrow \begin{bmatrix} A_p R + B_{pu} N_{A21} & A_p + B_{pu} N_{A22} C_{py} \\ N_{A11} & S A_p + N_{A12} C_{py} \end{bmatrix}$$

- Terms in blue are some of the transformed variables making up $\hat{\theta}$.

Projection

Let $\hat{\Theta}(J_{\Delta_p}, X)$ be the set of $\hat{\theta}$ which satisfy the LMI.

$$\min_{\hat{\theta}} \|\hat{\theta} - \hat{\theta}'\|_F$$

$$\text{s.t.}\, \hat{\theta} \in \hat{\Theta}\left(J_{\Delta_p}, X\right)$$

Take any controller $\hat{\theta}$ and find a similar one which guarantees closed-loop dissipativity.

Training

$$K^* = \arg\max_K \quad \mathbb{E}\left[\int_0^T r(x(t), u(t))dt\right]$$
$$\text{s.t.} \quad K \text{ makes closed-loop dissipative}$$

General Idea

Alternate between:

- Reinforcement learning step to improve controller
- Projection step to ensure dissipativity

Training Alg #1

Basic training in $\hat{\theta}$ space.

$\hat{\theta} \leftarrow$ random in Θ
while not converged **do**
  $\hat{\theta}' \leftarrow$ gradient step from $\hat{\theta}$
  $\hat{\theta} \leftarrow \arg\min_{\hat{\theta}} \|\hat{\theta} - \hat{\theta}'\|_F$ s.t. $\mathrm{LMI}(\hat{\theta})$
end while
$\tilde{\theta} \leftarrow f(\hat{\theta})$ $\triangleright$ Recover $\tilde{\theta}$

Training Alg #2

Training in θ space.

- In practice, works better than training in $\hat{\theta}$ space.

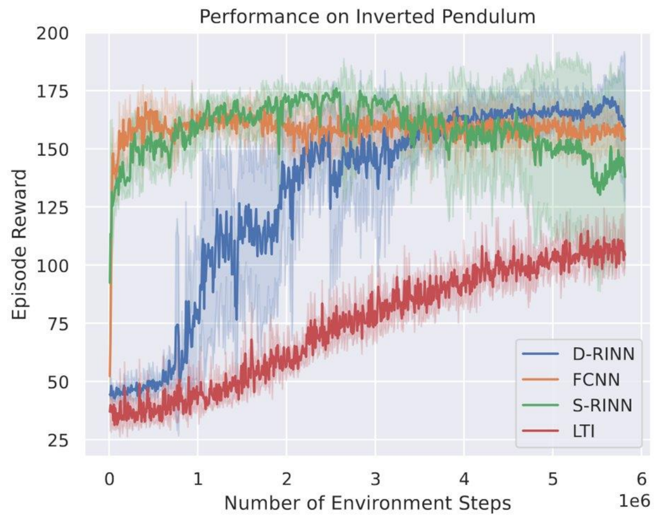
$\theta \leftarrow$ arbitrary
$P, \Lambda \leftarrow I$
for $i = 1, \ldots$ **do**
▷ Reinforcement learning step ◁
    $\theta' \leftarrow \textsc{ReinforcementLearningStep}(\theta)$
▷ Dissipativity-enforcing step ◁
    if $\exists P', \Lambda' : \theta'$ is dissipative **then**
        $\theta, P, \Lambda \leftarrow \theta', P', \Lambda'$
    else
        $\hat{\theta}' \leftarrow \textsc{ConstructThetaHat}(\theta', P, \Lambda)$
        $\hat{\theta} \leftarrow \textsc{ThetaHatProject}(\hat{\theta}', \hat{\Theta}(J_{\Delta_p}, X))$
        $P, \Lambda \leftarrow \textsc{ExtractFrom}(\hat{\theta})$
        $\theta \leftarrow \arg\min_{\theta} \|\theta - \theta'\| : \theta \in \Theta(J_{\Delta_p}, X, P, \Lambda)$
    end if
end for

Experiment 1: Inverted Pendulum

- Stabilize inverted pendulum with minimal control effort.

$$\dot{\boldsymbol{x}}_1(t) = \boldsymbol{x}_2(t)$$
$$\dot{\boldsymbol{x}}_2(t) = -\frac{\mu}{m\ell^2}\boldsymbol{x}_2(t) + \frac{g}{\ell}\sin(\boldsymbol{x}_1(t)) + \frac{1}{m\ell^2}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{x}_1(t)$$

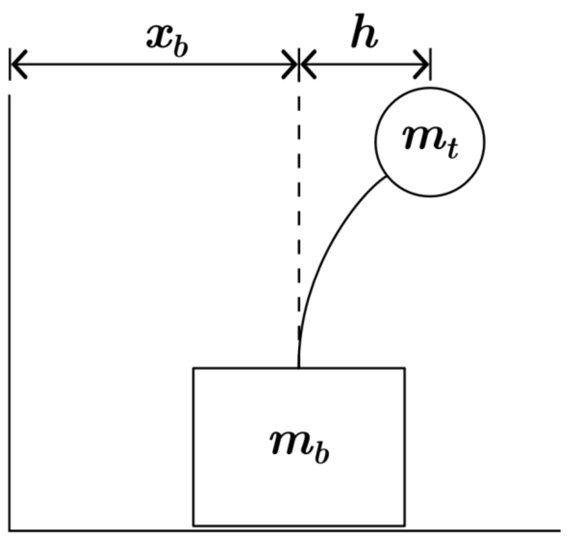
- Model this with $\Delta_p(x_1) = \sin(x_1)$ and $J_{\Delta_p} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$.
  - This is a sector-bound of [0,1] which holds over $[-\pi, \pi]$.



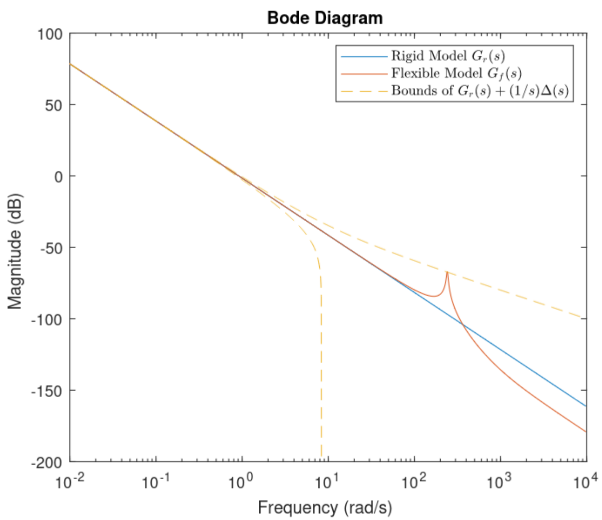
- D-RINN: Our method.
- FCNN: Fully connected NN.
- S-RINN: RINN without dissipativity constraints.
- LTI: LTI controller with dissipativity constraints, trained with our method.

Experiment 2: Flexible Rod on a Cart

- No joint; rod is flexible.
- Design with simplified model that assumes rod is rigid, with uncertainty to capture the difference between the rigid and flexible models.
- Train with flexible model to minimize state norm and control effort.

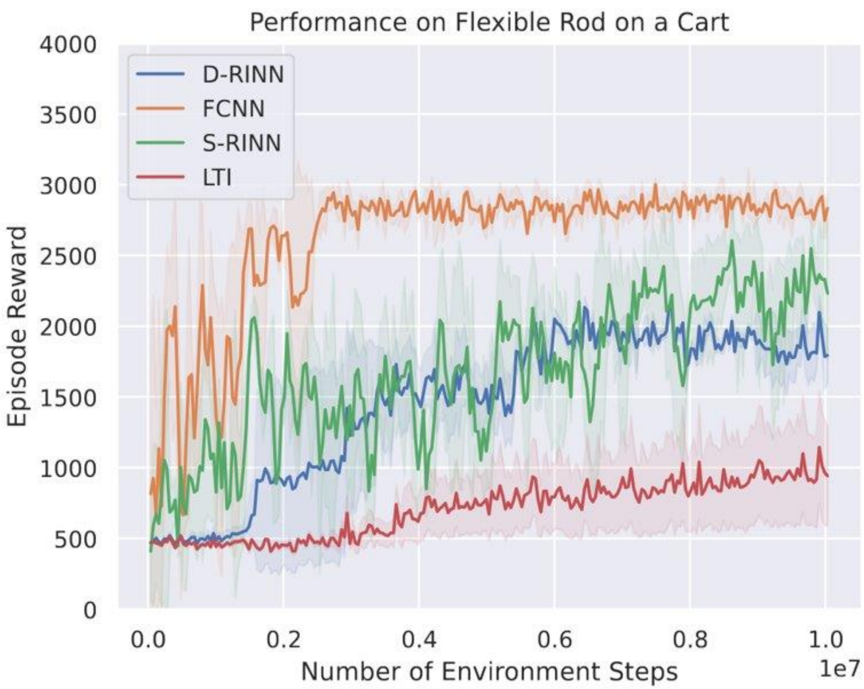


Bound uncertainty
with $\|\Delta(s)\| \leqslant 0.1$.



Experiment 2: Flexible Rod on a Cart

- L_2 gain constraint.
- Train to minimize control effort and state norm.



- D-RINN: Our method.
- FCNN: Fully connected NN.
- S-RINN: RINN without dissipativity constraints.
- LTI: LTI controller with dissipativity constraints, trained with our method.

Training: Issues

- Training recurrent policies
  - Vanishing gradients, slow training
- Conditioning of solution to projection

Training: Ill-Conditioned Solutions

θ is the set of controller parameters.

$\hat{\theta}$ is the set of variables in which the dissipation inequality is convex.

- Large gains in θ quickly result in nans in rollouts.
- Primary cause: Projection of $\hat{\theta}'$ into safe set results in ill-conditioned P in $\hat{\theta}$.

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \succ 0$$

R *and* S *parameterize* P

Training: Fixes to Ill-Conditioned Solutions

1. Backoff: allow some suboptimality in solution.

$$\delta^* \triangleq \min_{\hat{\theta}} \|\hat{\theta} - \hat{\theta}'\| \quad \text{s.t.} \begin{bmatrix} R & I \\ I & S \end{bmatrix} \succ 0, \ldots$$

$$\hat{\theta} \triangleq \arg\max_{\hat{\theta}, \epsilon} \epsilon \quad \text{s.t.} \begin{bmatrix} R & I \\ I & S \end{bmatrix} \succ \epsilon I, \|\hat{\theta} - \hat{\theta}'\| \leqslant \beta\delta^*, \ldots$$

2. Select t experimentally and use:

$$\begin{bmatrix} R & tI \\ tI & S \end{bmatrix} \succ 0$$

Training: Implementation Notes

- PyTorch and RLLib for learning framework
- Proximal Policy Optimization (PPO) for the RL algorithm
- CVXPY and Mosek for solving SDPs

Differential-Algebraic Equations (DAEs)

Dissipativity of DAEs

The dynamical model now has algebraic constraints:

$$y \leftarrow \boxed{\begin{aligned} \dot{x} &= f(x,u,z) \\ y &= h(x,u,z) \\ 0 &= g(x,u,z) \end{aligned}} \leftarrow u$$

If we can solve for z as a function of x, u from $g(x,u,z) = 0$, we get an ODE, but this elimination may be impractical (e.g., implicit NNs) or undesirable if it destroys useful structure (e.g., power networks).

Assume f, g, h vanish when $(x,u,z) = (0,0,\bar{z})$ for some $\bar{z}$.

The system above is dissipative with supply rate $s(u,y)$ if there exist $\lambda \geq 0$ and positive semidefinite $V : \mathbb{R}^n \to \mathbb{R}$ such that

$$\nabla V(x)^\top f(x,u,z) \leq s(u, h(x,u,z)) + \lambda \|g(x,u,z)\|^2 \quad \forall x,u,z$$

Note the algebraic constraint implies: $\dfrac{d}{dt} V(x(t)) \leq s(u(t), y(t))$

Dissipativity of DAEs

Example: Linear system $\dot{x} = Ax + B_u u + B_z z$

$$y = Cx + D_u u + D_z z$$

$$0 = Fx + G_u u + G_z z$$

Take quadratic storage function $V(x) = x^\top P x$:

$$\nabla V(x)^\top (Ax + B_u u + B_z z) = \begin{bmatrix} x \\ u \\ z \end{bmatrix}^\top \begin{bmatrix} A^\top P + PA & PB_u & PB_z \\ B_u^\top P & 0 & 0 \\ B_z^\top P & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ z \end{bmatrix}$$

and quadratic supply rate:

$$s(u, y) = \begin{bmatrix} u \\ y \end{bmatrix}^\top X \begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} x \\ u \\ z \end{bmatrix}^\top \begin{bmatrix} 0 & I & 0 \\ C & D_u & D_z \end{bmatrix}^\top X \begin{bmatrix} 0 & I & 0 \\ C & D_u & D_z \end{bmatrix} \begin{bmatrix} x \\ u \\ z \end{bmatrix}$$

Then dissipation inequality becomes LMI:

$$-\begin{bmatrix} A^\top P + PA & PB_u & PB_z \\ B_u^\top P & 0 & 0 \\ B_z^\top P & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & C^\top \\ I & D_u^\top \\ 0 & D_z^\top \end{bmatrix} X \begin{bmatrix} 0 & I & 0 \\ C & D_u & D_z \end{bmatrix} + \lambda \begin{bmatrix} F^\top \\ G_u^\top \\ G_z^\top \end{bmatrix} \begin{bmatrix} F & G_u & G_z \end{bmatrix} \succeq 0$$

Dissipativity of DAEs

SOS formulation: For polynomial f, g, h, s look for polynomial V s.t.

$$V(x) - \epsilon x^\top x \in \Sigma[x]$$

$$s(u, h(x,u,z)) + \lambda g(x,u,z)^\top g(x,u,z) - \nabla V(x)^\top f(x,u,z) \in \Sigma[x,u,z]$$

Special case: Take $s(u,y) = 0$ and $\epsilon > 0$ to prove stability of the origin in the absence of input.

Example:

$$\dot{x}_1 = -x_1 + z$$

$$\dot{x}_2 = -x_1 - x_2$$

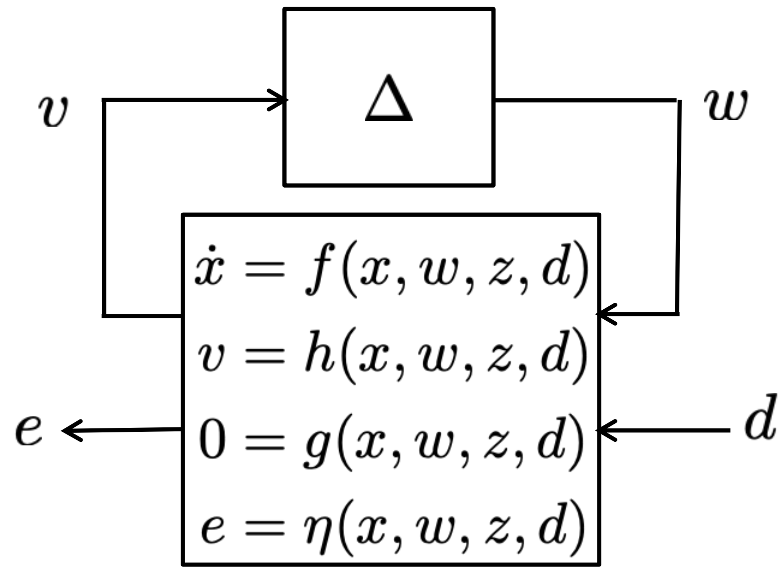
$$0 = x_1^2 + (x_2^2 + 5)z$$

When we allow V be polynomial of degree 4 and let $\epsilon = 10^{-3}$ SOSTOOLS and SeDuMi find $\lambda = 0.59504$ and

$$\begin{aligned} V(x) = {} & 0.00017634 x_1^4 + 0.0012261 x_1^2 x_2^2 + 0.0027498 x_1 x_2^3 \\ & + 0.0023039 x_2^4 + 0.013246 x_1^3 - 0.013733 x_1^2 x_2 - 0.055089 x_1 x_2^2 \\ & - 0.056305 x_2^3 + 0.40316 x_1^2 + 0.67688 x_1 x_2 + 0.57717 x_2^2 \end{aligned}$$

Dissipativity of DAEs

Robust Stability/Performance:



Performance objective: disipativity with supply rate $\sigma(d, e)$.

Stability: special case with $\sigma(d, e) = 0$ and positive definite, not just semidefinite, storage function.

If Δ satisfies quadratic constraints $\begin{bmatrix} v \\ w \end{bmatrix}^\top J_k \begin{bmatrix} v \\ w \end{bmatrix} \geq 0, k = 1, 2, \ldots$

look for $\lambda \geq 0,\ \tau_k \geq 0$ and positive semidef. V s.t. for all x, w, z, d

$$\nabla V(x)^\top f(x, w, z, d) \leq \sigma(d, e) + \underbrace{\lambda \| g(x, w, z, d)) \|^2}_{=0} \underbrace{- \textstyle\sum_k \tau_k \begin{bmatrix} v \\ w \end{bmatrix}^\top J_k \begin{bmatrix} v \\ w \end{bmatrix}}_{\leq 0}$$

Dissipativity of DAEs

Example: Power Network

Analyze performance of a wide-area controller under line failures

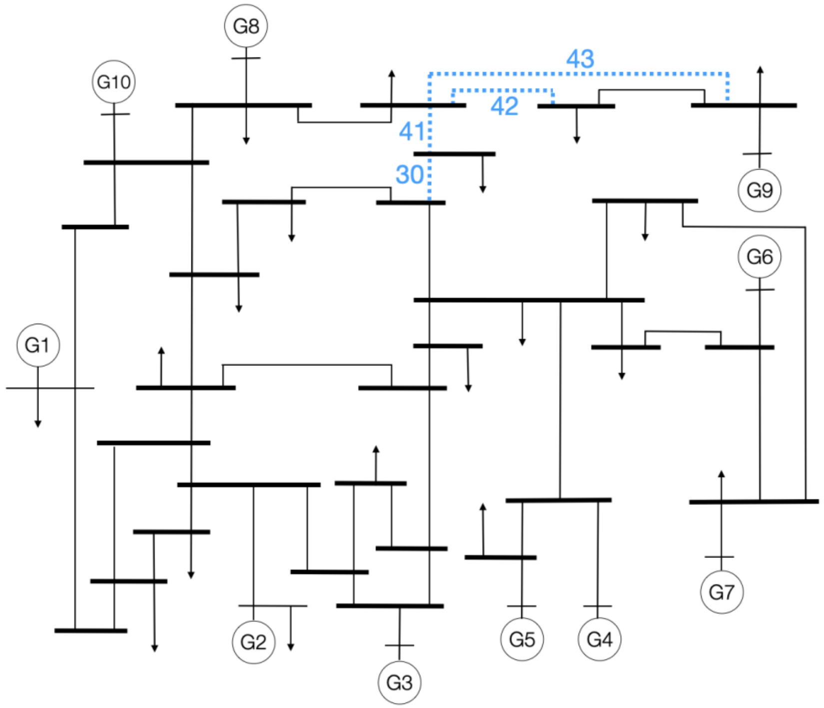
Swing equations and power flow equations linearized about power flow solution:

$$\frac{d}{dt}\begin{bmatrix}\tilde{\delta}\\ \tilde{\omega}\end{bmatrix} = \overline{A}\begin{bmatrix}\tilde{\delta}\\ \tilde{\omega}\end{bmatrix} + \overline{B}_z z + \begin{bmatrix}0\\ u+d\end{bmatrix}$$

$$0 = \overline{F}\begin{bmatrix}\tilde{\delta}\\ \tilde{\omega}\end{bmatrix} + Gz$$

$\tilde{\delta}, \tilde{\omega}$: deviation from set point of angle, angular velocity vectors

u, d : control and disturbance



IEEE 39-Bus network. Blue dashed lines: potential line failures

DAE model avoids inversion of poorly conditioned G and retains the network structure embedded in G.

Dissipativity of DAEs

Example: Power Network

Define reduced state $x = Q \begin{bmatrix} \tilde{\delta} \\ \tilde{\omega} \end{bmatrix}$ to eliminate rotational symmetry.

Columns of Q form orthonormal basis $\perp \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}$

Model incorporating state feedback controller:

$$\dot{x} = A_{cl}x + B_z z + B_d d$$
$$0 = Fx + Gz$$
$$e = Cx$$

A group of potential line failures (whose effect on power flow sol'n is negligible) can be captured with polytopic model replacing G with:

$$G_0 + \sum_i \theta_i \underbrace{K_i L_i^\top}, \;\; \theta_i \in [-1, 1]$$

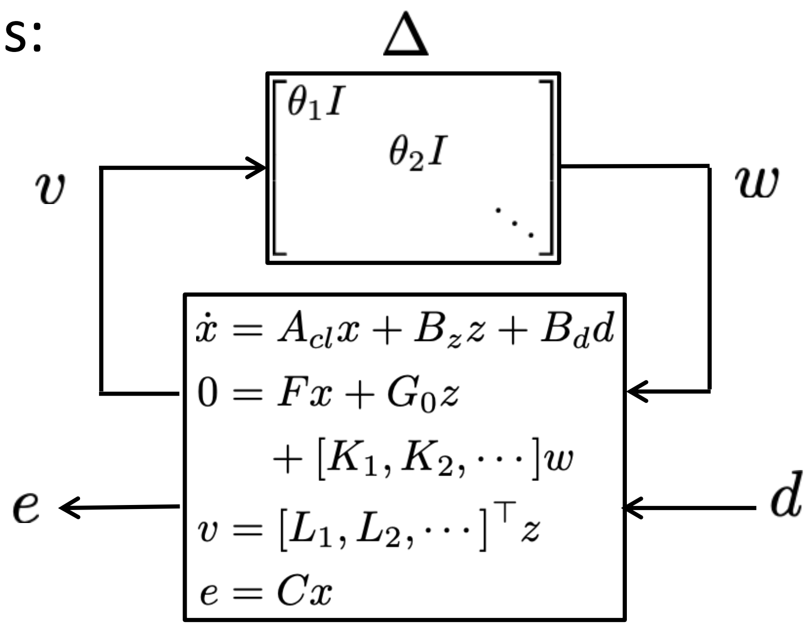
Low-rank perturbation from failure i

K_i, L_i : tall matrices

Dissipativity of DAEs

Example: Power Network

Represent model as:



Δ satisfies the quadratic constraint:

$$\begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} X & Y \\ Y^\top & -X \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \geq 0$$

for any block diagonal X, Y where the blocks X_i, Y_i conform to the sizes of identity multiplying θ_i, and $Y_i = -Y_i^\top, X_i = X_i^\top \succeq 0$

Dissipativity of DAEs

Dissipation inequality for performance:

$$\underbrace{\nabla V(x)^\top f(x,w,z,d)}_{①} \leq \sigma(d,e) + \lambda \underbrace{\|g(x,w,z,d))\|^2}_{②} - \underbrace{\textstyle\sum_k \tau_k \begin{bmatrix} v \\ w \end{bmatrix}^\top J_k \begin{bmatrix} v \\ w \end{bmatrix}}_{③}$$

$$① \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix}^\top \begin{bmatrix} A^\top P + PA & 0 & PB_z & PB_d \\ 0 & 0 & 0 & 0 \\ B_z^\top P & 0 & 0 & 0 \\ B_d^\top P & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix} \qquad ② \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix}^\top \begin{bmatrix} F^\top \\ K^\top \\ G_0^\top \\ 0 \end{bmatrix} \begin{bmatrix} F & K & G_0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix}$$

$$③ \begin{bmatrix} v \\ w \end{bmatrix}^\top \begin{bmatrix} X & Y \\ Y^\top & -X \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix}^\top \begin{bmatrix} 0 & 0 \\ 0 & I \\ L^\top & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X & Y \\ Y^\top & -X \end{bmatrix} \begin{bmatrix} 0 & 0 & L & 0 \\ 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ z \\ d \end{bmatrix}$$

If $\sigma(d,e)$ quadratic, e.g., $\sigma(d,e) = \gamma^2\|d\|^2 - \|e\|^2$ for L_2 gain γ, we can write dissipation inequality above as LMI in decision variables P, λ, X, Y, where X, Y constrained as in previous slide. We can also let γ be a decision variable and make it the objective to minimize.

Dissipativity of DAEs

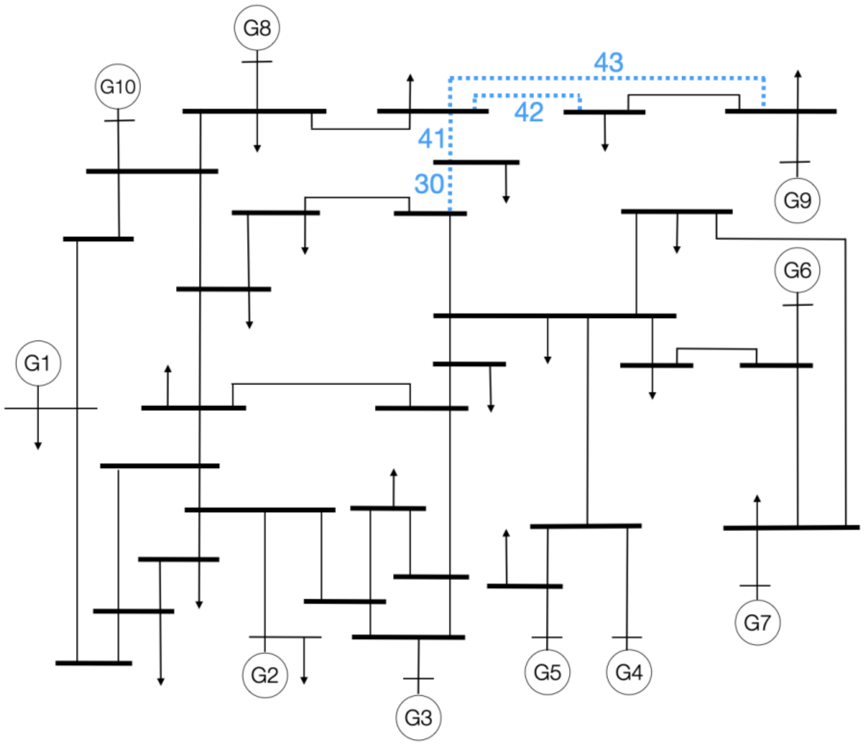
Example: Power Network

Analyze performance of a wide-area controller under line failures

Procedure in previous slide applied to the IEEE 39-bus with a wide-area controller. LMI finds L_2 gain 2.31 over the uncertainty set related to failure of lines 30, 41, 42, 43.

Not a conservative estimate. L_2 gains computed for individual line removals:

Line removed	30	41	42	43
Closed-loop H_∞-norm	2.215	2.222	2.219	2.217



IEEE 39-Bus network. Blue dashed lines: potential line failures

For details see Jensen et. al, arXiv:2308.08471

Further Reading

- Willems, Brockett, Some new rearrangement inequalities having application in stability analysis, IEEE TAC, 1968.
- Willems, The analysis of feedback systems, M.I.T. Press, 1971.
- Scherer, Gahinet, Chilali, Multiobjective output-feedback control via LMI optimization, TAC, 1997.
- Fazlyab, Robey, Hassani, Morari, Pappas, Efficient and accurate estimation of Lipschitz constants for deep neural networks, NeurIPS, 2019.
- Fazlyab, Morari, Pappas, Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming, TAC, 2020.
- Pauli, Koch, Berberich, Kohler, Allgöwer, Training robust neural networks using Lipschitz bounds, IEEE CSL, 2021.
- Yin, Seiler, Arcak, Stability analysis using quadratic constraints for systems with neural network controllers, TAC, 2021.
- Yin, Seiler, Jin, Arcak, Imitation learning with stability and safety guarantees, IEEE CSL, 2021.

Further Reading

- Ebihara, Waki, Magron, Mai, Peaucelle, Tarbouriech, ℓ_2 induced norm analysis of discrete-time LTI systems for nonnegative input signals and its application to stability analysis of recurrent neural networks," EJC 2021.
- Pauli, Gramlich, Berberich, Allgöwer, Linear systems with neural network nonlinearities: Improved stability analysis via acausal ZamesFalb multipliers, CDC, 2021.
- Pauli, Funcke, Gramlich, Msalmi, Allgöwer, Neural network training under semidefinite constraints, CDC, 2022.
- Junnarkar, Yin, Gu, Arcak, Seiler, Synthesis of stabilizing recurrent equilibrium network controllers, CDC 2022.
- Gu, Yin, El Ghaoui, Arcak, Seiler, Jin, Recurrent neural network controllers synthesis with stability guarantees for partially observed systems, AAAI, 2022.
- Revay, Wang, Manchester, Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness, IEEE TAC, 2023.
- Wang, Barbara, Revay, Manchester, Learning over all stabilizing nonlinear controllers for a partially-observed linear system, IEEE CSL, 2023.

Further Reading

- Richardson, Turner, Gunn, Strengthened circle and Popov criteria for the stability analysis of feedback systems with ReLU neural networks, IEEE CSL, 2023.
- Junnarkar, Arcak, Seiler. Synthesizing Neural Network Controllers with Closed-Loop Dissipativity Guarantees, arXiv.
- Vahedi Noori, Hu, Dullerud, Seiler, Stability and Performance Analysis of Discrete-Time ReLU Recurrent Neural Networks, arXiv.
- Jensen, Junnarkar, Arcak, Wu, Gumussoy. Certifying Stability and Performance of Uncertain Differential-Algebraic Systems: A Dissipativity Framework, arXiv.