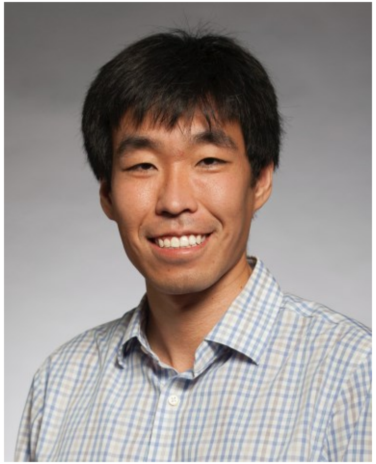
Structured Learning for Control of Networked Systems : Stability and Steady-State Tracking Guarantees

Wenqi Cui

2023.12.05

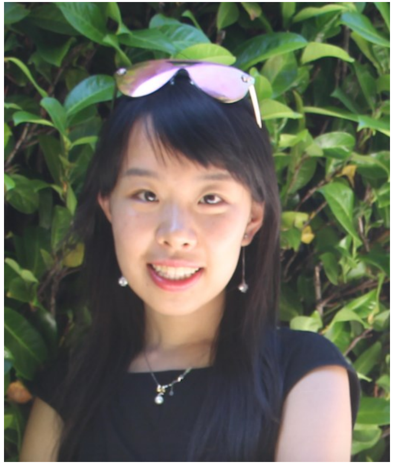
ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

Acknowledgements



Prof. Baosen Zhang
University of Washington

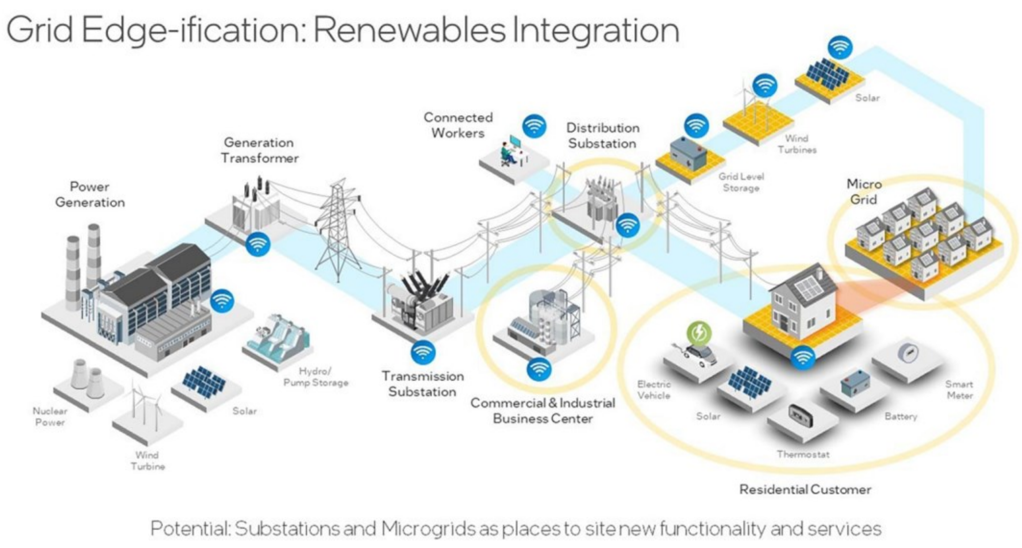
Dr. Yan Jiang
University of Washington

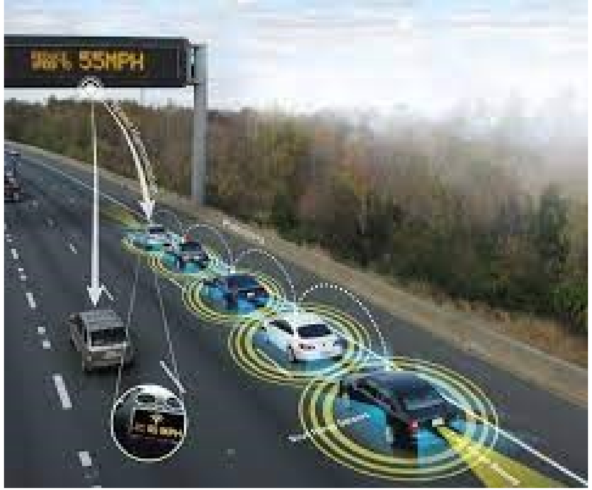


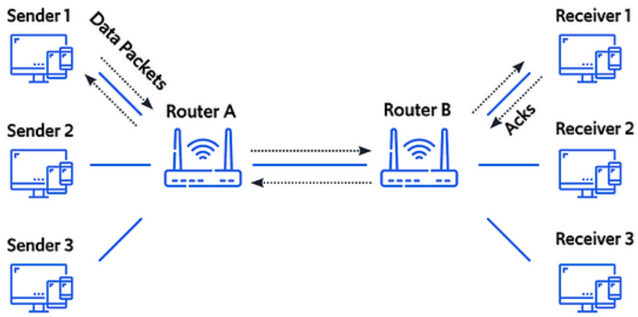
Prof. Yuanyuan Shi
University of California
San Diego

Prof. Jorge Cortes
University of California
San Diego

1 Networked Systems



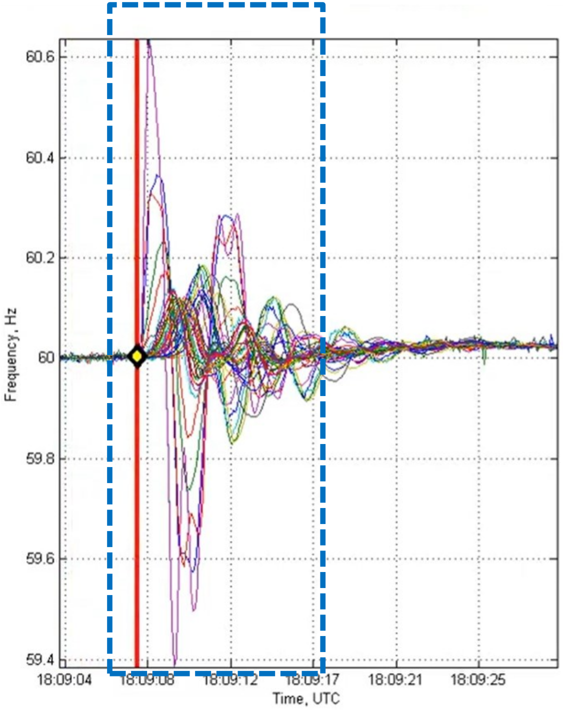


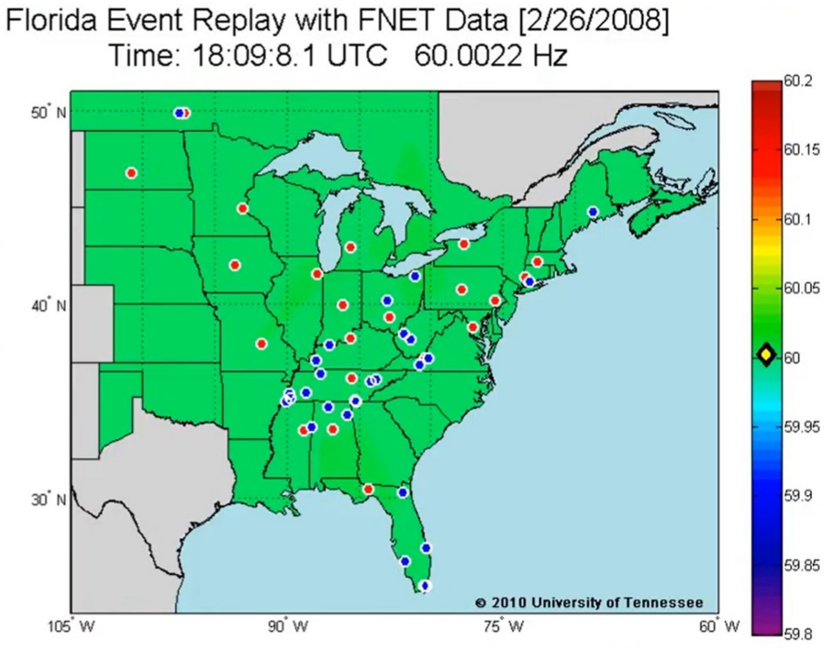


https://www.intel.com/content/www/us/en/newsroom/opinion/tech-can-solve-grids-climate-change-challenge.html

1 Performance metrics

Transient Period: Reduce the largest frequency deviations





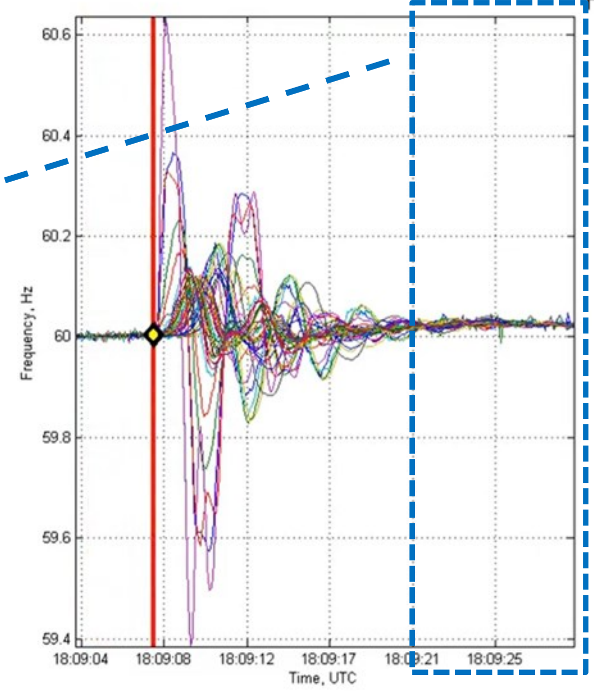
THE UNIVERSITY of TENNESSEE UT OAK RIDGE National Laboratory CURENT

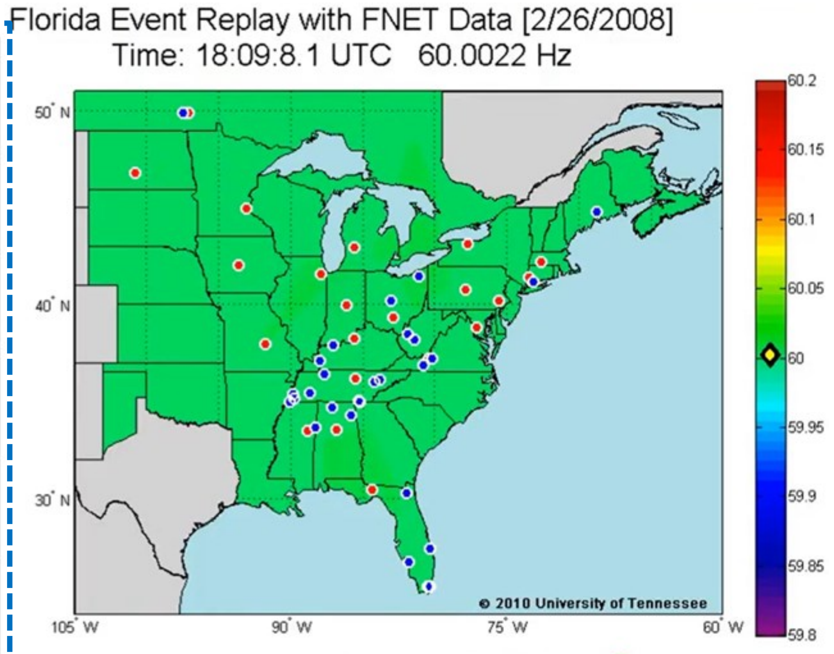
https://www.youtube.com/watch?v=H7y-oJYpDkM

1 Performance metrics

Steady state:

- ❑ Frequency reaches agreement at 60Hz



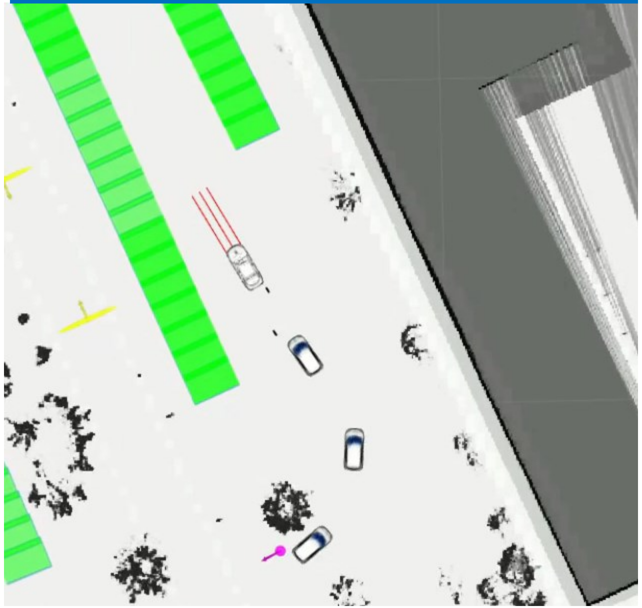


THE UNIVERSITY of TENNESSEE UT OAK RIDGE National Laboratory CURENT

https://www.youtube.com/watch?v=H7y-oJYpDkM

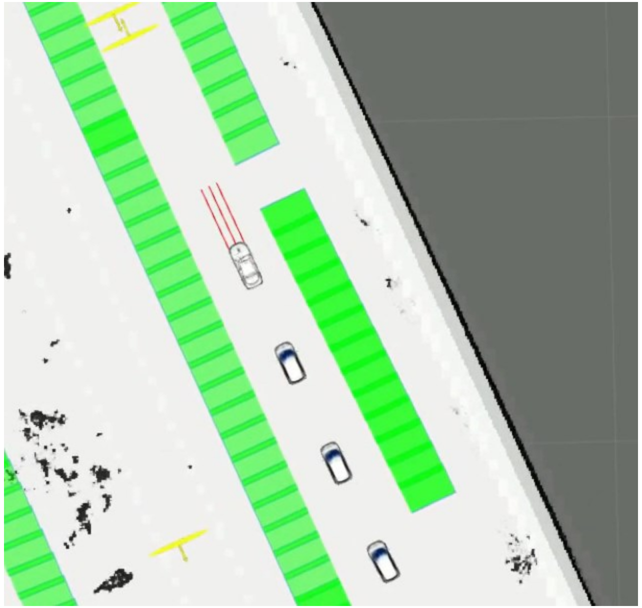
1 Performance metrics

Transient period performance



- ❑ Keep the relative distance

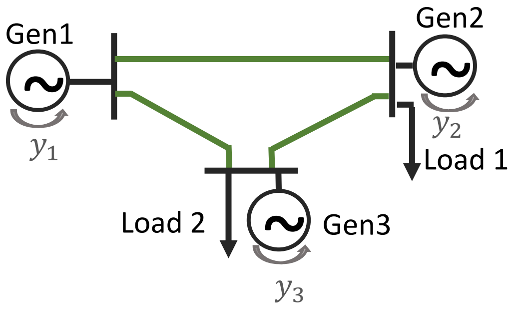
Steady-state performance



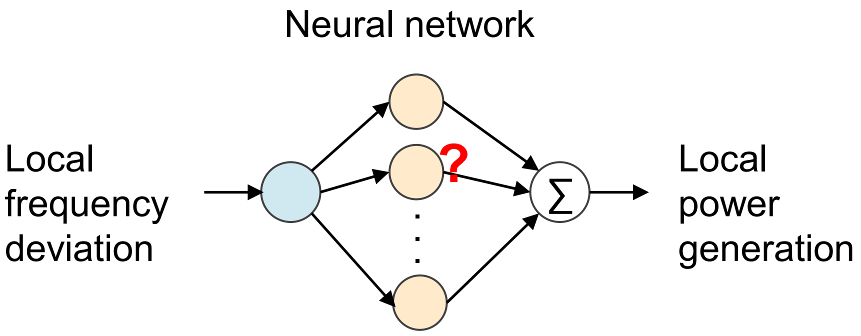
- ❑ The desired velocity

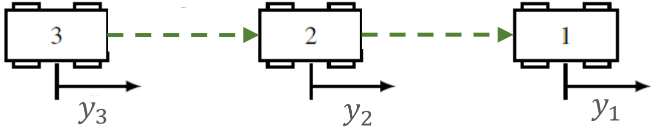
https://www.youtube.com/watch?v=F90K-rYQRAY

1 Learning for Control

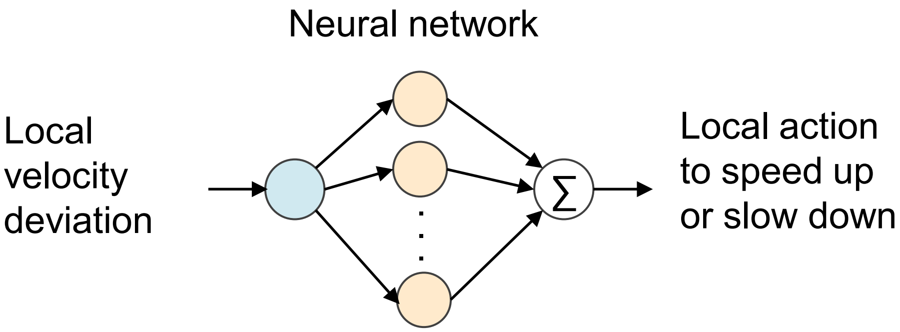


(a) Example 1: A power grid





(b) Example 2: A vehicle platoon



1 Learning for Control

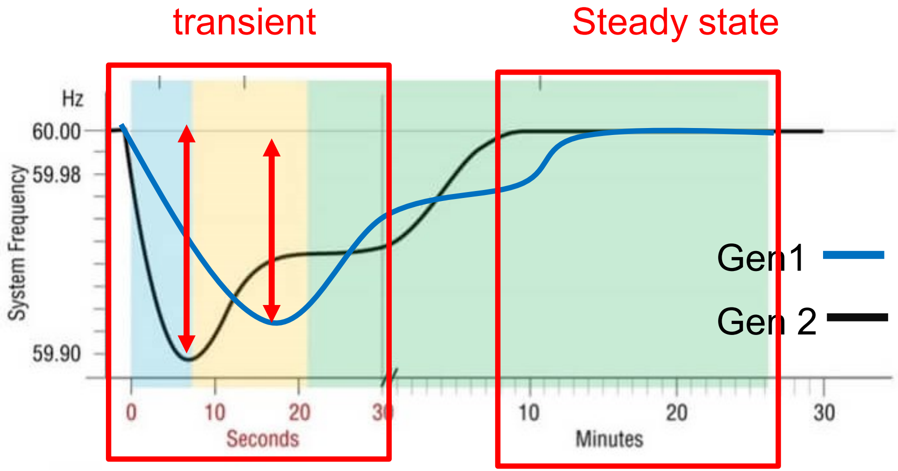
Transient Optimization

Training neural networks

$$\min_{u} \sum_{t=0}^{T} \textit{Transient Cost}$$

$$s.t. \quad \textit{dynamics of the system}$$

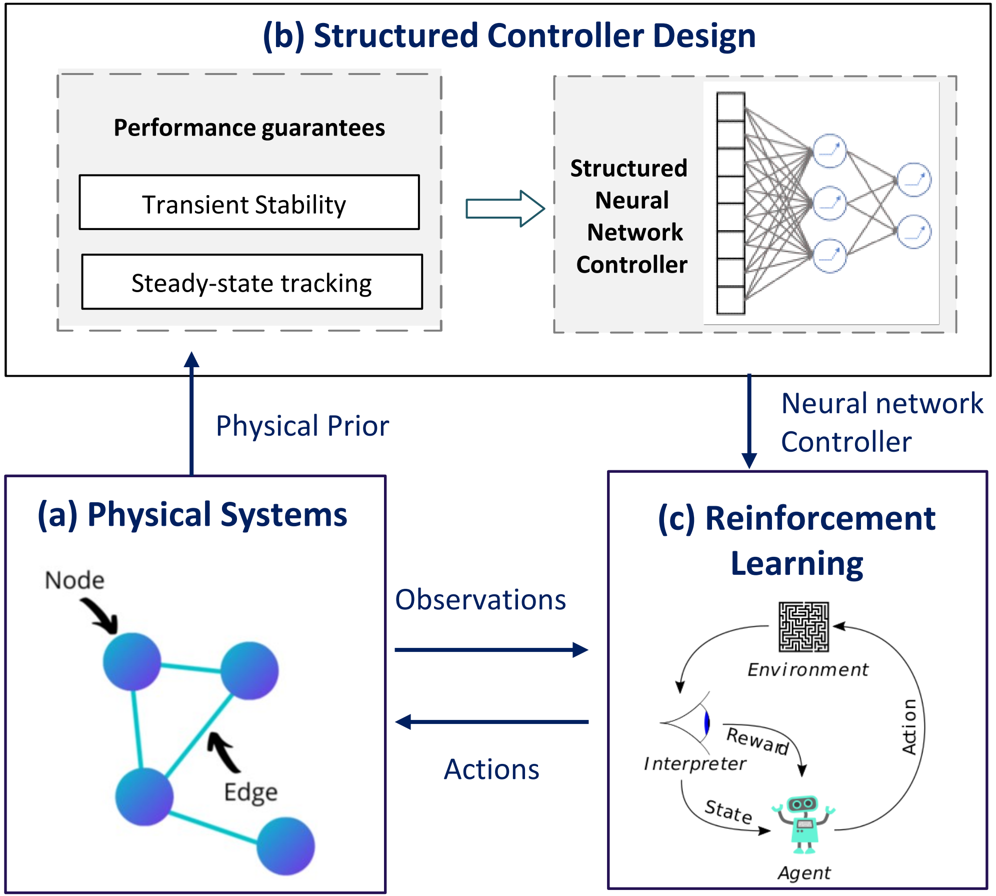
$$\textit{transient stability and output tracking}$$



Provable guarantees for a range of tracking point?

1 Key idea

- ❑ Derive structural properties of controllers satisfying performance guarantees
- ❑ Enforce the structures in the design of neural networks



2 Power System Transient Dynamics

The swing equation for power system transient dynamics

Angle

Frequency deviation

nonlinear neural networks

Local static controller that change active power for frequency control

$$\dot{\delta}_i = \omega_i,$$

$$M_i\dot{\omega}_i = -D_i\omega_i - d_i - u_i\,(\omega_i) - \sum_{(i.j)\in\mathcal{E}} \underbrace{B_{ij}\sin(\delta_i - \delta_j)}_{\text{Power Flow}}$$

Inertia constant

Damping coefficient

Net power extraction

The susceptance of the line (i, j)

2 Transient and Steady-state Optimization

Transient performance

$$\min_{u(\omega)} \sum_{i=1}^{n} \sum_{t=0}^{T} J_i\left(\omega_i(t), u_i(\omega_i(t))\right)$$

$$s.t. \qquad dynamics\ of\ the\ system$$

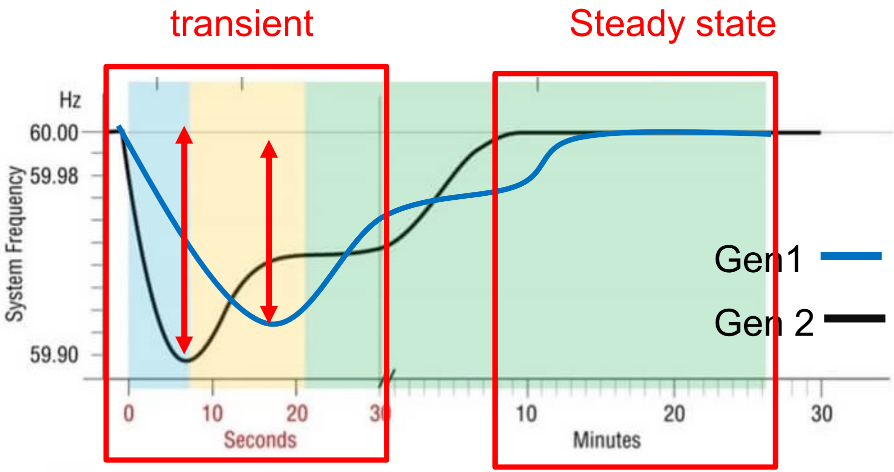
$$\underline{a}_i \le a_i(\omega_i(t)) \le \overline{a}_i$$

$$\boxed{a_i(\omega_i(t)) \text{ is stabilizing}}$$

Steady-state performance

- Frequency Restoration

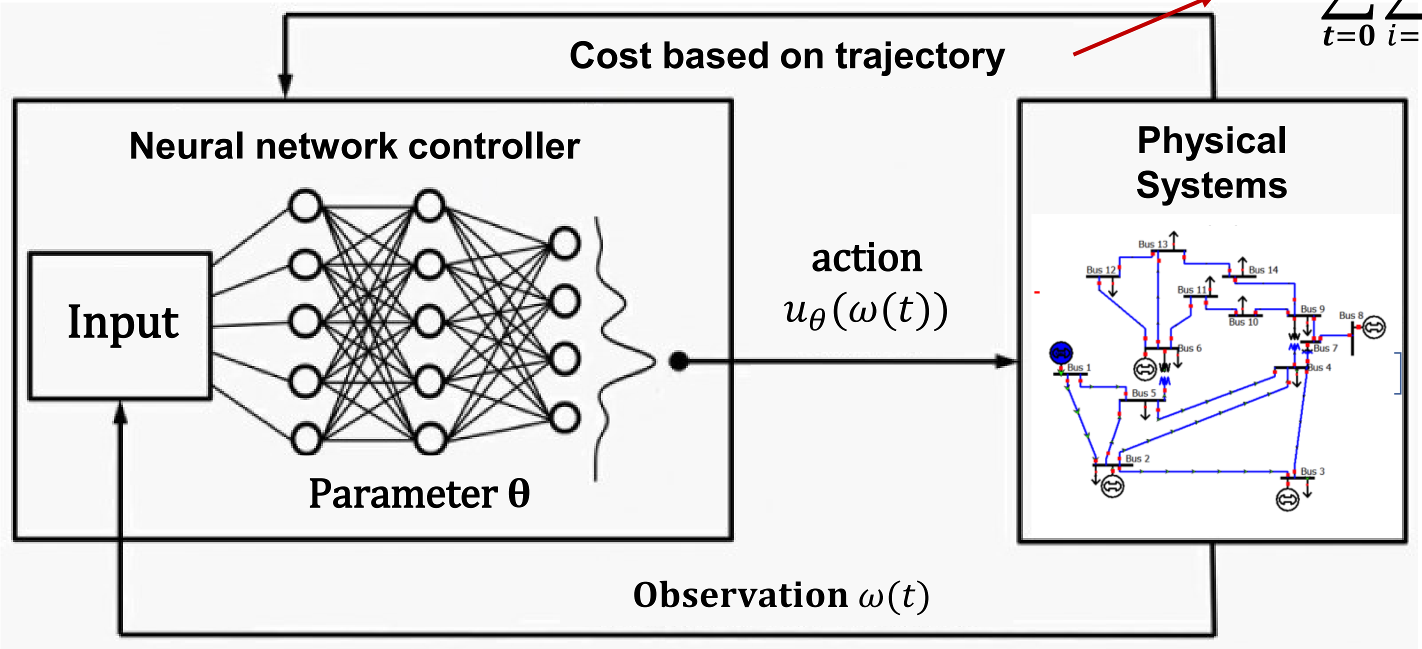
$$\omega_i^* = 0$$



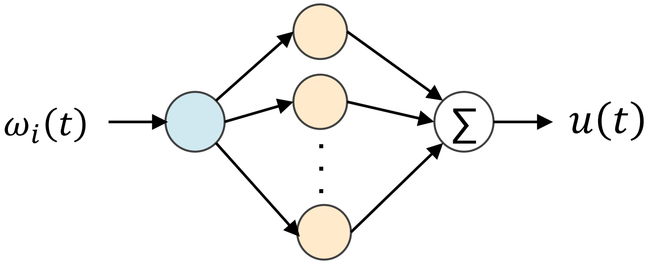
2 Transient Optimization with RL

- Parameter the controller with neural network controller
- Reinforcement learning (RL) for training the neural network

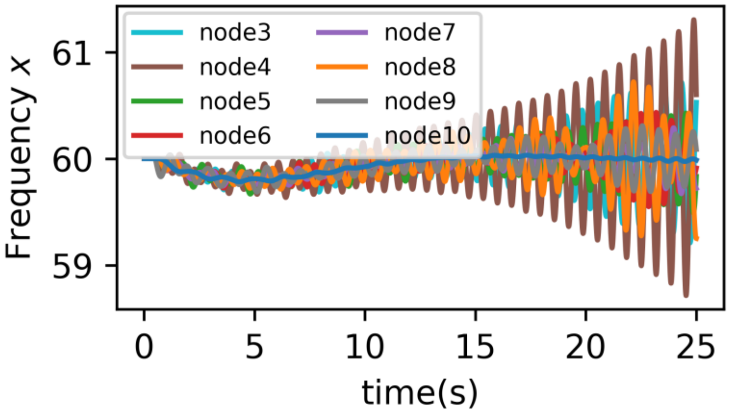
$$\boldsymbol{Loss} = \sum_{t=0}^{T} \sum_{i=1}^{n} J_i(\omega_i(t), u_i(t))$$



2 Hard Constraint on Stability



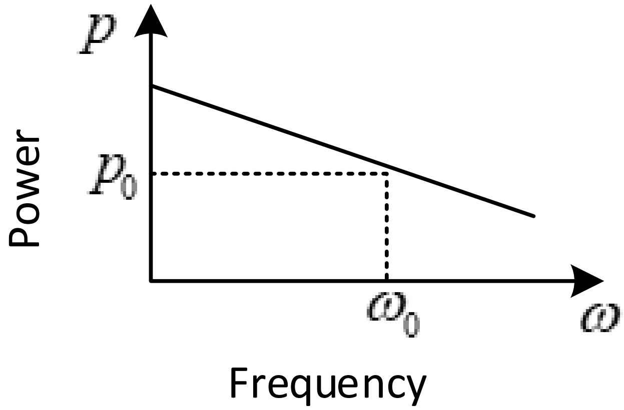
$$\boldsymbol{Loss} = \sum_{t=0}^{T} \sum_{i=1}^{n} J_i(\omega_i(t), u_i(t))$$



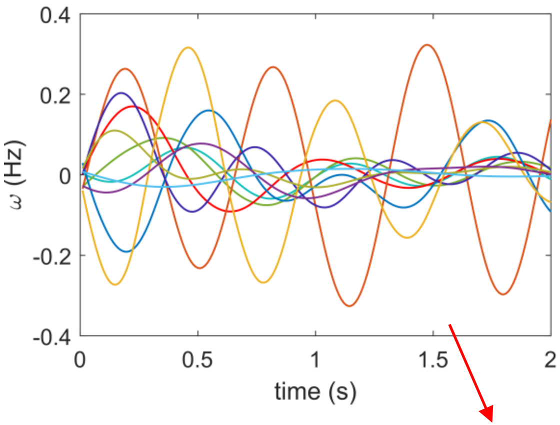
(a) Dynamics for system with generic neural network controllers

2 Hard Constraint on Stability

Current approaches in literature



Option 1 : linearized control law and system dynamics



Option 2 : RL without stability requirement, or simply add soft penalties

2 Lyapunov Approach for a Stabilizing Controller

A local Lyapunov function $V(\delta, \omega)$ for the dynamic system is

$$V(\boldsymbol{\delta}, \boldsymbol{\omega}) = \sum_{(i.j)\in\mathcal{E}} \underbrace{-B_{ij}\cos(\delta_{ij}) + B_{ij}\cos(\delta_{ij}^*) - B_{ij}\sin(\delta_{ij}^*)(\delta_{ij} - \delta_{ij}^*)}_{\text{Potential Energy (Bregman distance of } -B_{ij}\cos(\delta_{ij}))} + \frac{1}{2}\sum_{i=1}^{n} \underbrace{M_i(\omega_i - \omega_i^*)^2}_{\text{Kinetic Energy}}$$

It is a valid Lyapunov function in $\Theta = \left\{(\boldsymbol{\delta}, \boldsymbol{\omega}) | \delta_{ij} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \forall i.j\right\}$, where $-B_{ij}\cos(\delta_{ij})$ is convex and therefore

$$-B_{ij}\cos(\delta_{ij}) \geq \left(-B_{ij}\cos(\delta_{ij}^*)\right) + \left(-\nabla B_{ij}\cos(\delta_{ij}^*)\right)(\delta_{ij} - \delta_{ij}^*)$$

with equality holding if and only if $\delta_{ij} = \delta_{ij}^*$.

2 Lyapunov Approach for a Stabilizing Controller

Sufficient condition for asymptotic stability

$$\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) = \sum_{i=1}^{n} \frac{\partial V}{\partial \delta_i} \dot{\delta}_i + \frac{\partial V}{\partial \omega_i} \dot{\omega}_i$$

Independent with system parameter and topologies!

$$= \sum_{i=1}^{n} -D_i(\omega_i - \omega_i^*)^2 \boxed{- (\omega_i - \omega_i^*)(u_i(\omega_i) - u_i(\omega_i^*))}$$

$$\leq 0$$

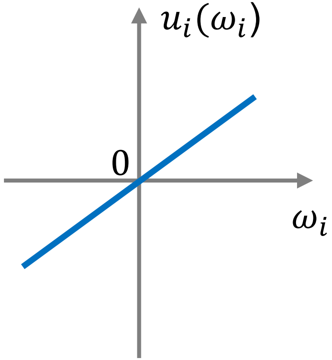
The condition $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) \leq 0$ holds if

$$(\omega_i - \omega_i^*)(u_i(\omega_i) - u_i(\omega_i^*)) \geq 0 \qquad \forall i = 1, \cdots n$$

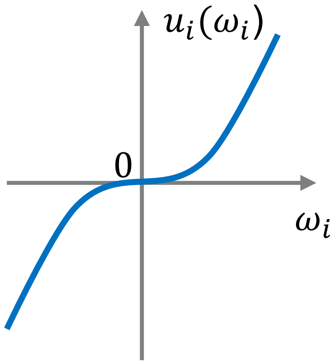
A unique equilibrium if $u_i(\omega_i)$ is monotonic increasing and cross the origin

2 Structural Property for a Stabilizing Controller

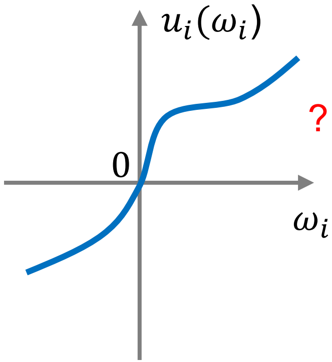
Monotonic increasing functions cross the origin



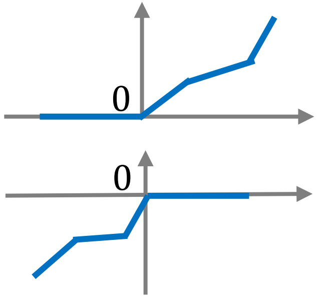
Linear



Polynomial



General form?



Neural Network?

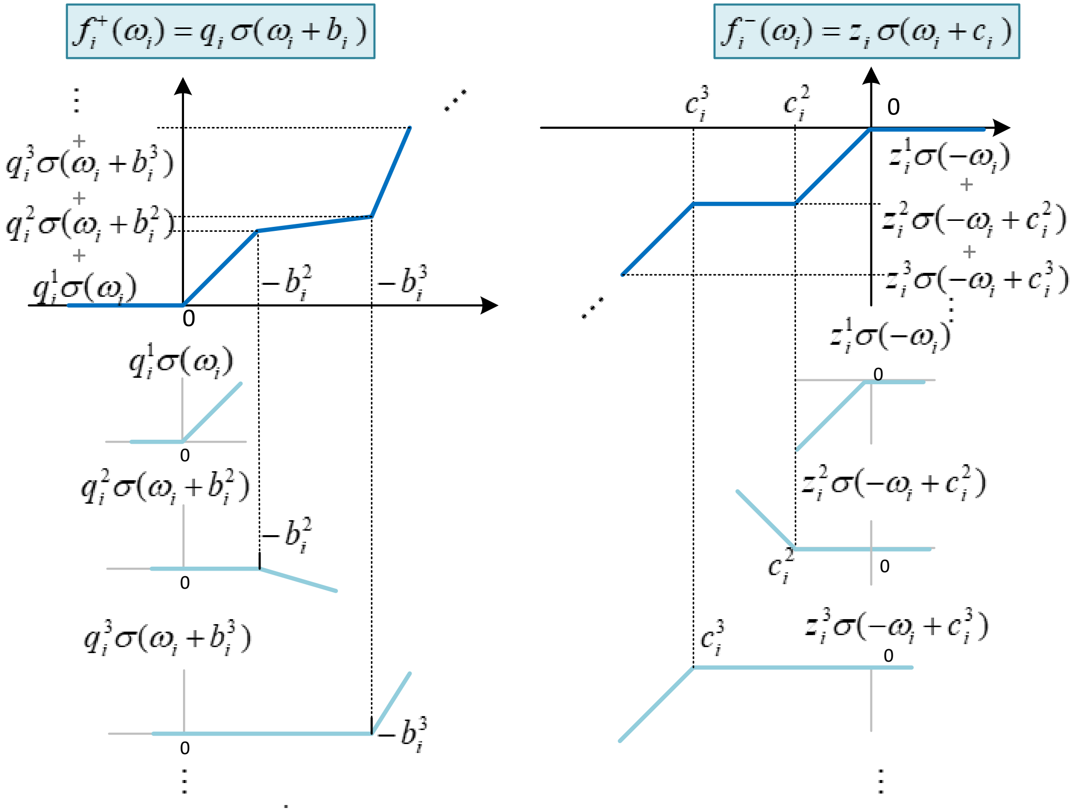
2 Structured Neural Network Design

ReLU function

$$\sigma(x) = \max(x, 0)$$

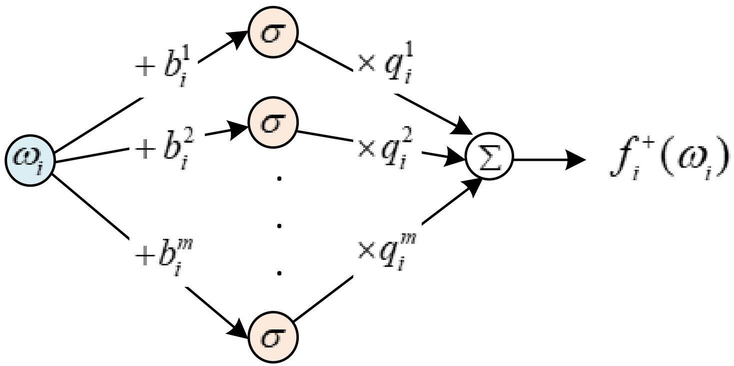
$$q\sigma(x+b) = \begin{cases} q(x+b) & if \;\; x > -b \\ 0 & otherwise \end{cases}$$

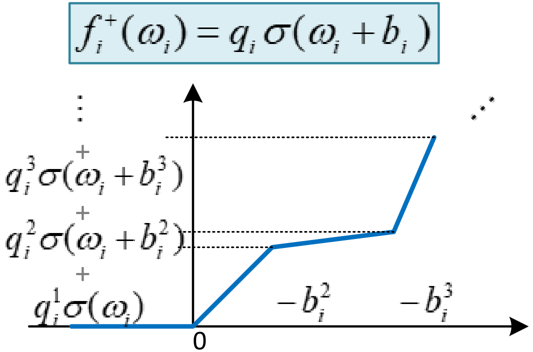
Explicitly engineer the structure of neural network controllers by stacking the ReLU function



2 Structured Neural Network Design

Structured one-layer neural network for parameterizing controller





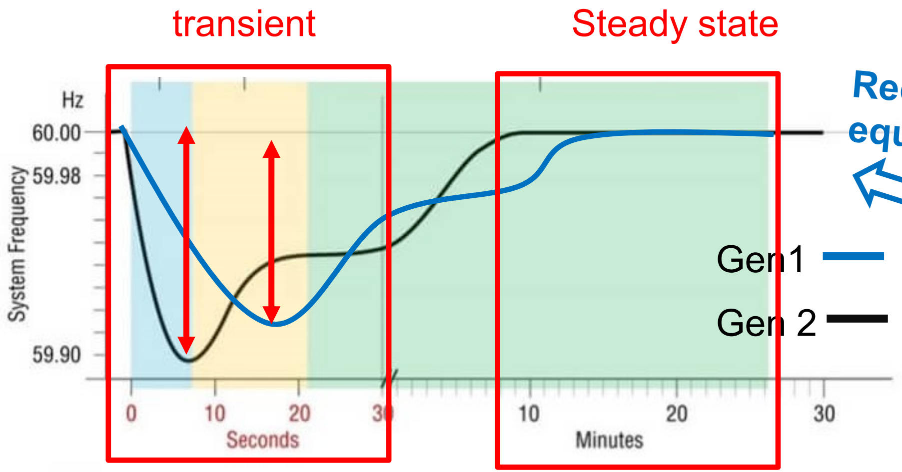
$$f_i^+(\omega_i) = \boldsymbol{q}_i\sigma(\mathbf{1}\omega_i + \boldsymbol{b}_i)$$

$$where \sum_{j=1}^{l} q_i^j \geq 0, \quad \forall l = 1, 2, \cdots, m$$

$$b_i^1 = 0, b_i^l \leq b_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m$$

We proved that $u_i(\omega_i) = f_i^+(\omega_i) + f_i^-(\omega_i)$ is a universal approximation of any bounded, Lipschitz continuous and monotonically increasing function through the origin.

3 Steady-State Tracking



Required equilibrium

Steady-state metrics

- ❑ Frequency Restoration

$$\omega_i^* = 0$$

- ❑ Structural property of controllers stabilize towards the required equilibrium
- ❑ Engineer the neural networks to satisfy these structures

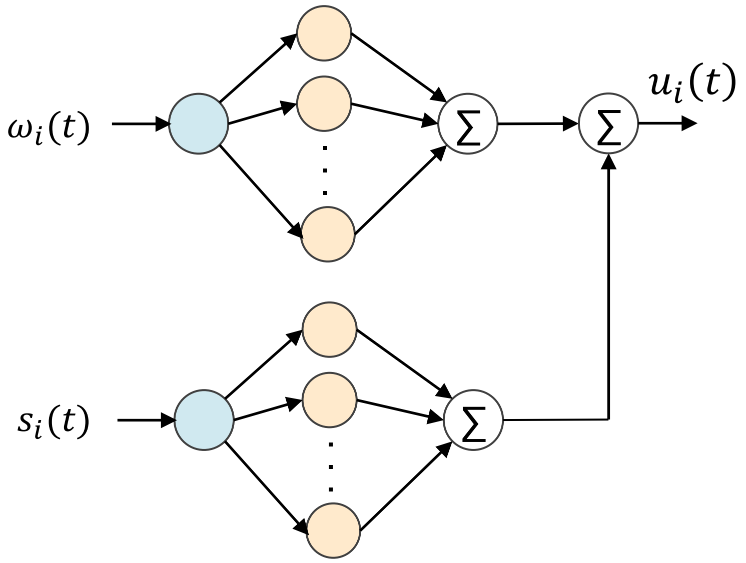
Satisfy by design

3 Control theoretic motivations

- Output agreement to the required value

Integral Control

$$\dot{s}_i = \overline{y} - y_i \quad \Longleftrightarrow \quad s_i(t) = \int_{\tau=0}^{t} \left(\overline{y} - y_i(\tau)\right) d\tau$$



3 Generalized PI Controller

$\dot{s}_i = \omega_i$

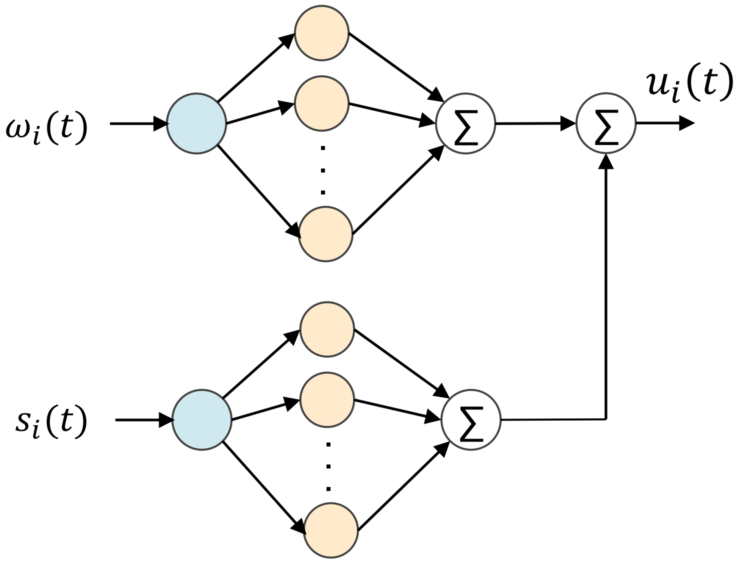
$u_i(t) = \pi_i^P(\omega_i) + \pi_i^I\big(s_i(t)\big)$

Generalized proportional control

Generalized integral control

Check equilibrium: If $\omega_i^* = \overline{y}$, then $\dot{s}_i = 0$

Whether trajectories from any possible initial states converges to this equilibrium?



3 Monotonicity →Convexity

Define the integral function $L_i(s_i) = \int_0^{s_i} \pi_i^I(z)dz$, which is strictly convex since $\nabla^2 L_i(s_i) = \frac{d\pi_i^I(s_i)}{ds_i} > 0$ by strictly monotonicity.

A local Lyapunov function $V(\eta, \omega, s)$ for the dynamic system is

$$V(\eta,\omega,s) = \sum_{(i.j)\in\mathcal{E}} -B_{ij}\cos(\delta_{ij}) + B_{ij}\cos(\delta_{ij}^*) - B_{ij}\sin(\delta_{ij}^*)(\delta_{ij}-\delta_{ij}^*) + \frac{1}{2}\sum_{i=1}^{n} M_i(\omega_i-\omega_i^*)^2$$

$$+\sum_{i=1}^{n} \underbrace{L_i(s_i) - L_i(s_i^*) - \nabla L_i(s_i^*)(s_i - s_i^*)}_{\text{Bregman distance with } L_i(s_i)}$$

3 Convergence of ω_i

The time derivative of the energy function is

$$\dot{V}(\eta, \omega, s) = \sum_{i=1}^{n} \frac{\partial V}{\partial \omega_i} \dot{\omega}_i + \frac{\partial V}{\partial s_i} \dot{s}_i + \sum_{l=1}^{m} \frac{\partial V}{\partial \eta_l} \dot{\eta}_l$$

$$\leq \textstyle\sum_{i=1}^{n} -\rho_i (\omega_i - \omega_i^*)^2 + \left(\pi_i^P(\omega_i) - \pi_i^P(\omega_i^*)\right)(\omega_i - \omega_i^*)$$

$$\leq \textstyle\sum_{i=1}^{n} -\rho_i (\omega_i - \omega_i^*)^2 \quad \text{(by monotone of } \pi_i^P\text{)}$$

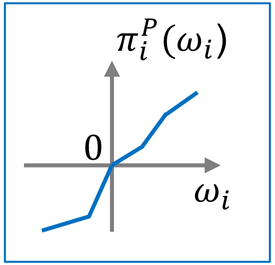
3 Generalized Neural-PI Controller with Guarantees

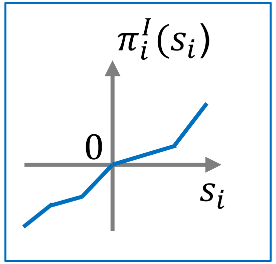
In summary, the control law given as follows stabilizes the system and also guarantees frequency restoration at the steady state

$$\dot{s}_i = \omega_i$$

$$u_i(\omega_i) = \pi_i^P(\omega_i) + \pi_i^I(s_i)$$

Monotone Neural Network Implementation



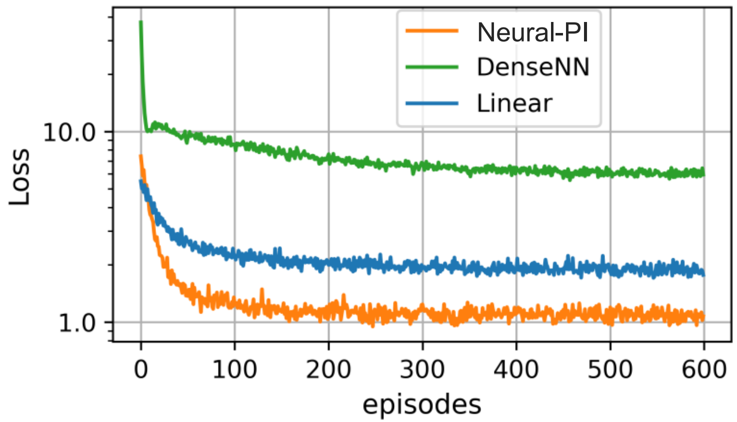


3 Case study : Power systems

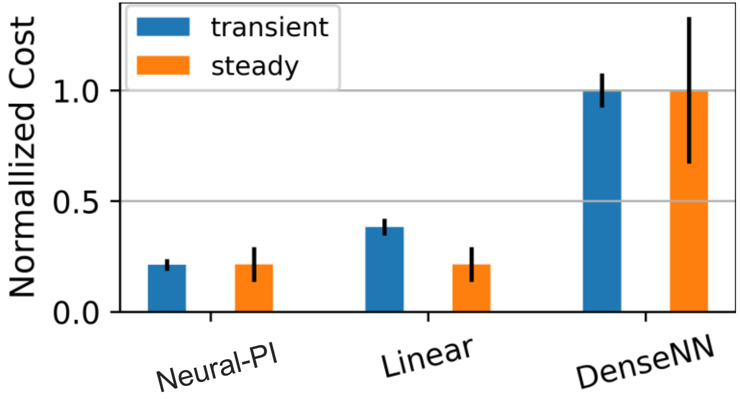
Case study is conducted on IEEE-39 bus test system.

Compare the proposed Neural-PI controller with

1) Linear PI control with the coefficient optimized by training
2) Dense neural network

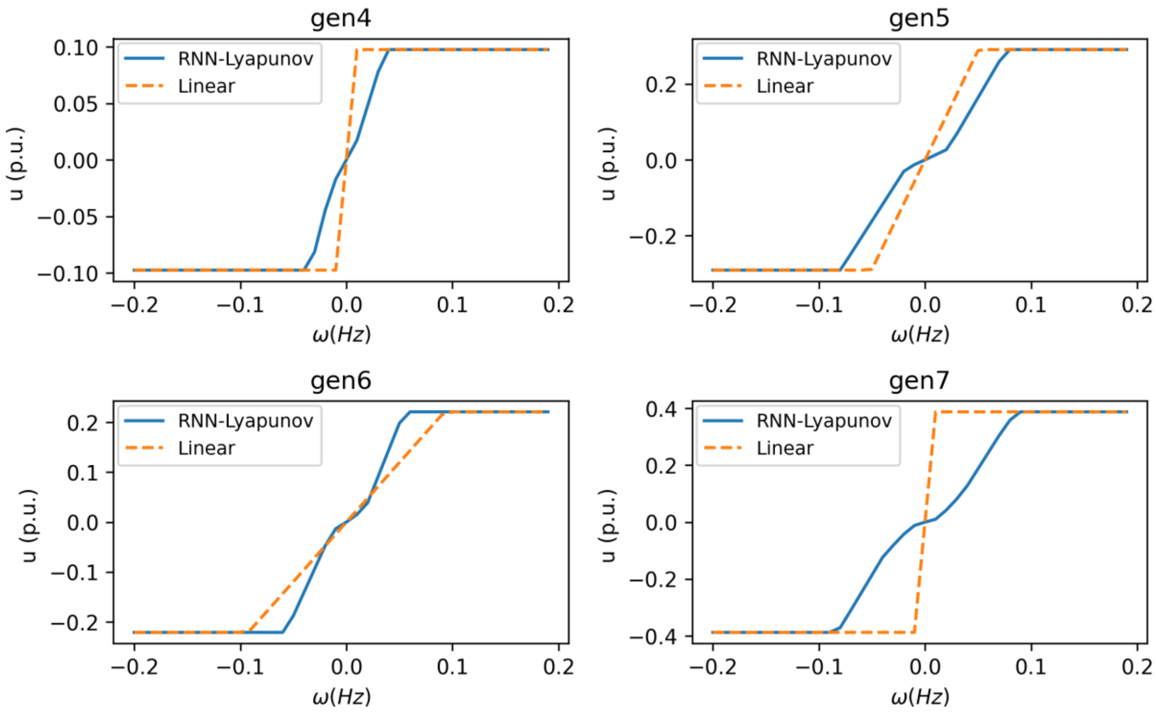


Training loss



Transient and steady-state cost

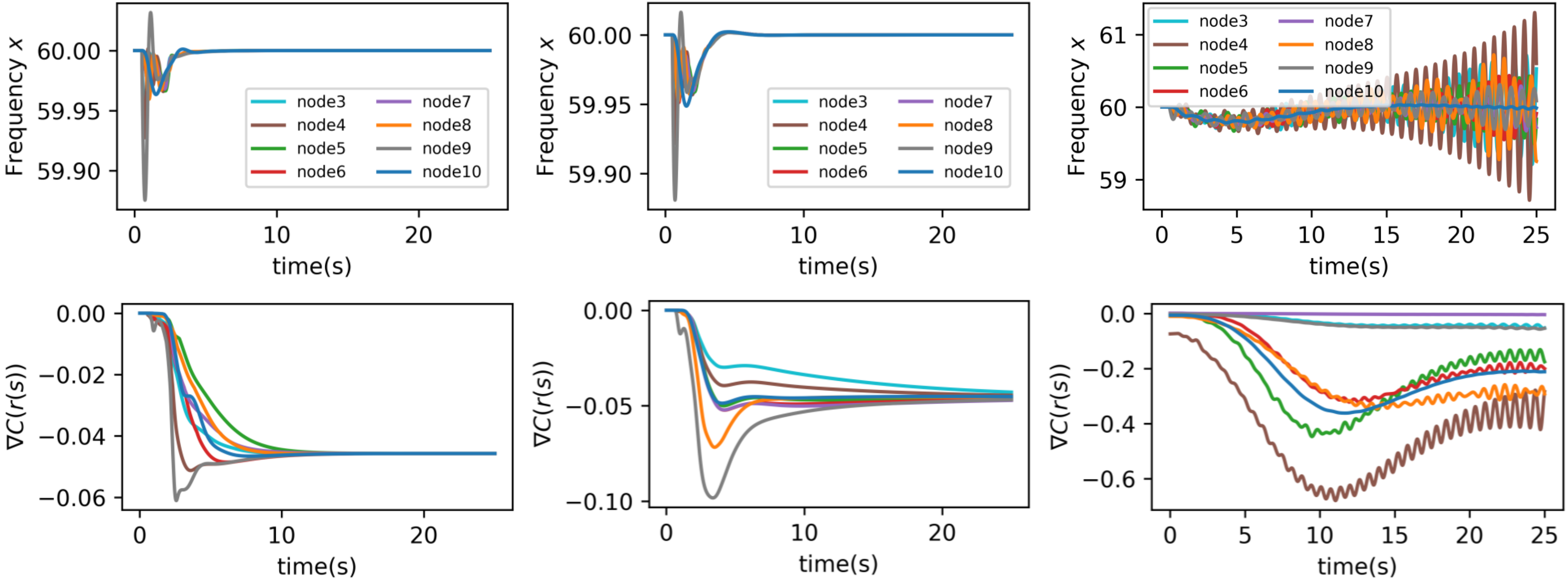
3 Case Study



Control Action *u* obtained by different approaches

3 Case study : Power systems

The dynamics after the same step load changes



(a) Neural-PI (b) Linear (c) DenseNN

4 Generalization to More Dynamic Systems

We consider a dynamic system described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}),\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}),$$

where state $\boldsymbol{x} \in \mathbb{R}^n$, output $\boldsymbol{y} \in \mathbb{R}^m$, control action $\boldsymbol{u} \in \mathbb{R}^m$.

Assumption: Equilibrium-Independent Passivity

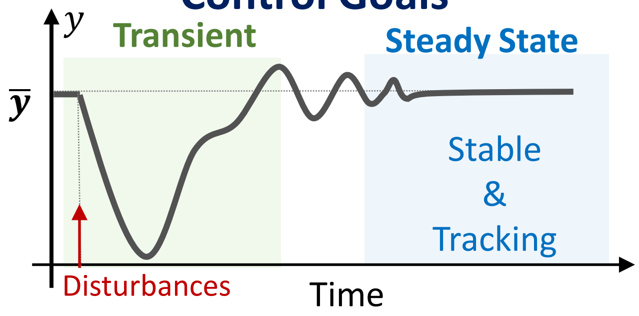
The system described by $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}),\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x})$ is strictly equilibrium-independent passive (EIP) if it satisfies:

(i) for every equilibrium $\boldsymbol{u}^*$, there exists a unique $\boldsymbol{x}^*$ such that $\boldsymbol{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) = \boldsymbol{0}$, and

(ii) there exists a positive definite storage function $S(\boldsymbol{x}, \boldsymbol{x}^*)$ a constant ρ such that

$$S(\boldsymbol{x}^*, \boldsymbol{x}^*) = 0 \ \text{ and } \ \dot{S}(\boldsymbol{x}, \boldsymbol{x}^*) \leq (\boldsymbol{y} - \boldsymbol{y}^*)^T(\boldsymbol{u} - \boldsymbol{u}^*) - \rho||\boldsymbol{y} - \boldsymbol{y}^*||^2$$

4 Generalized PI Controller



The control law is

$$\dot{s} = \bar{y} - y$$

$$u = \underbrace{\pi^P(\bar{y} - y)}_{\text{Proportional control}} + \underbrace{\pi^I(s)}_{\text{Integral control}}$$

Strictly Monotone Functions

Gradient of convex function $\nabla g(z)$ is strictly monotone

Definition: Strictly Monotone Functions

A continuous function $q\colon \mathbb{R}^m \mapsto \mathbb{R}^m$ is strictly monotone on $D \subset \mathbb{R}^m$ if $\left(q(\eta) - q(\xi)\right)^T(\eta - \xi) \geq 0,\ \forall \eta, \xi \in D$, with the equality holds if and only if $\eta = \xi$.

4 Stability

Lyapunov function := Storage function + Bregman Distance of Convex Functions

Comes from $\boldsymbol{\pi}^I(\boldsymbol{s}) = \nabla g(\boldsymbol{z})$

Definition: Bregman Distance

The Bregman Distance of a strictly convex function g: $\mathbb{R}^m \mapsto \mathbb{R}$ at the point s^* is defined as $B(\boldsymbol{s}, \boldsymbol{s}^*) = g(\boldsymbol{s}) - g(\boldsymbol{s}^*) - \left(\nabla g(\boldsymbol{s}^*)\right)^T(\boldsymbol{s} - \boldsymbol{s}^*)$, which is positive definite with equality holds only when $s = s^*$.

4 Stability

Lyapunov function := Storage function + Bregman Distance of Convex Functions

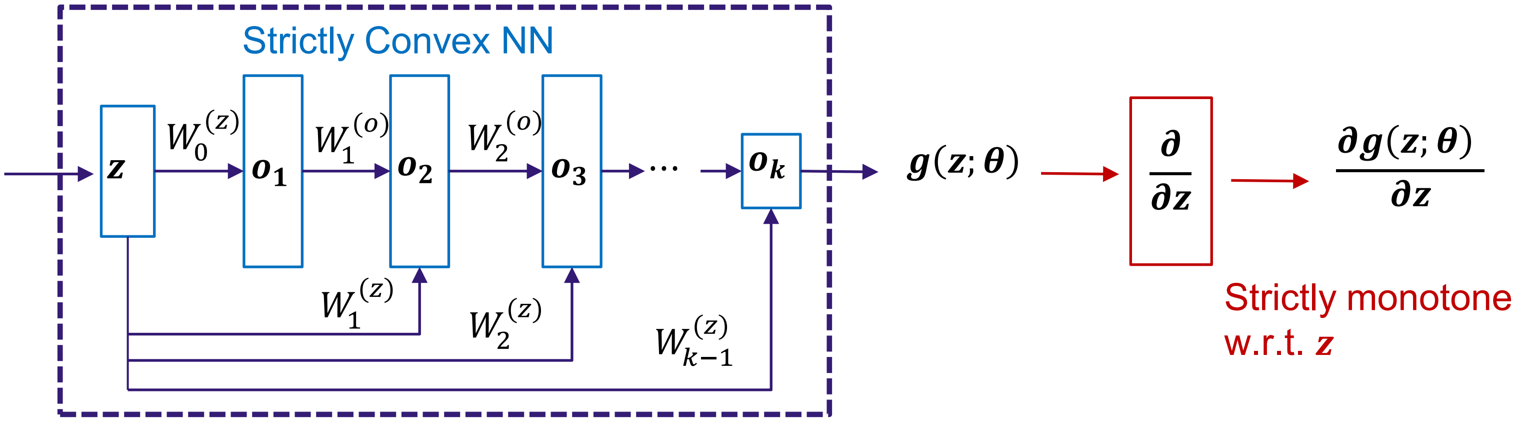
Comes from $\boldsymbol{\pi}^I(\boldsymbol{s}) = \nabla g(\boldsymbol{z})$

$$\dot{V} \leq -\rho||\boldsymbol{y} - \boldsymbol{y}^*||^2 + \underbrace{\boxed{(\boldsymbol{y} - \boldsymbol{y}^*)^T \left(\boldsymbol{\pi}^P(\overline{\boldsymbol{y}} - \boldsymbol{y}) - \boldsymbol{\pi}^P(\overline{\boldsymbol{y}} - \boldsymbol{y}^*)\right)}}_{\leq 0 \text{ by monotonicity}}$$

Definition: Bregman Distance

The Bregman Distance of a strictly convex function $g\colon \mathbb{R}^m \mapsto \mathbb{R}$ at the point s^* is defined as $B(\boldsymbol{s}, \boldsymbol{s}^*) = g(\boldsymbol{s}) - g(\boldsymbol{s}^*) - \left(\nabla g(\boldsymbol{s}^*)\right)^T (\boldsymbol{s} - \boldsymbol{s}^*)$, which is positive definite with equality holds only when $s = s^*$.

4 Strictly Convex NN $\rightarrow$ Monotone NN



A strictly convex function $g(z;\theta)$ parameterized by k -layer neural network, with o_l being the output of the l-th layer

$$o_{l+1} = \sigma_l\left(W_l^{(o)} o_l + W_l^{(z)} z + b_l\right), \qquad \mathrm{g}(\mathrm{z};\theta) = o_k$$

Strictly convex and increasing

Positive except for $o_0, W_0^{(o)} \equiv 0$

4 Strictly Convex NN - Universal approximation

Convex function ← Max of affine function ← DenseNN with ReLU ← DenseNN with softplus-beta

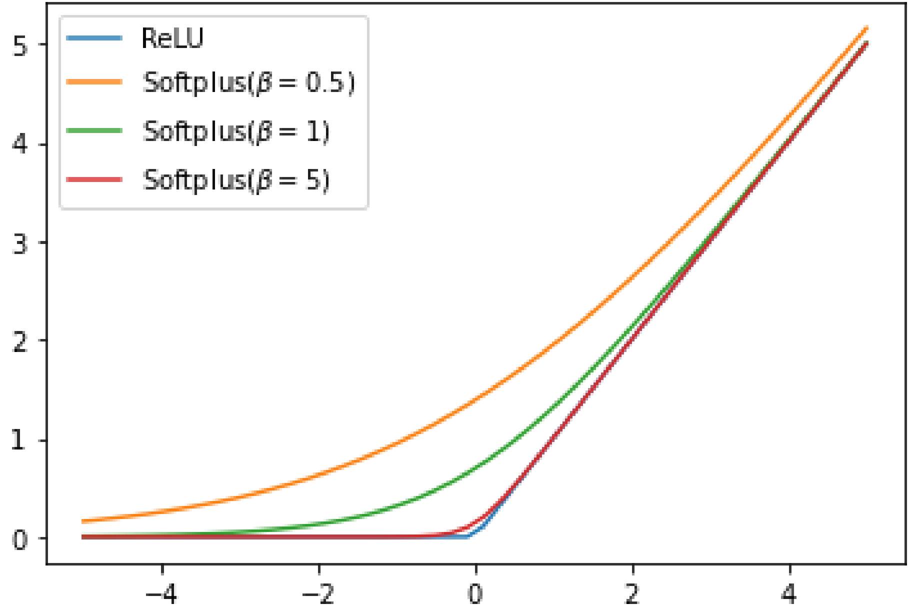
Universal approximation of any strictly convex functions

$$o_{l+1} = \sigma_l \left(W_l^{(o)} o_l + W_l^{(z)} z + b_l \right),$$

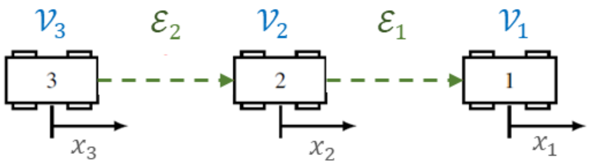
$$\mathrm{g}(\mathrm{z}; \theta) = o_k$$

when the activation is

$$\sigma_l^{softplus}(x) \coloneqq \frac{1}{\beta} \log\left(1 + e^{\beta x}\right)$$



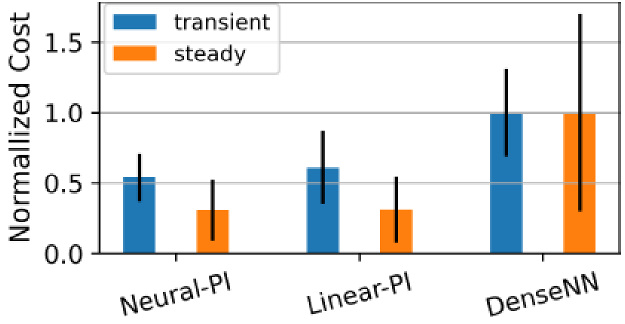
4 Case study : Vehicle platooning



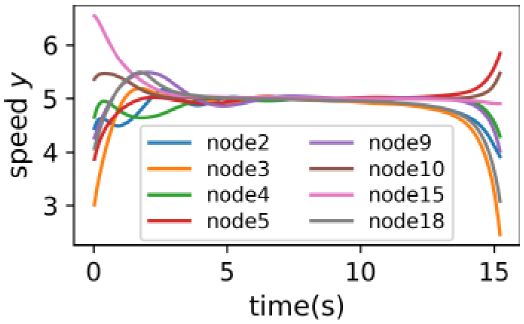
Vehicle platooning problem with 20 vehicles

Compare the proposed Neural-PI controller with

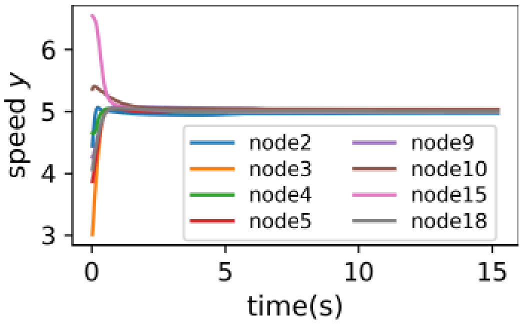
1) Linear PI control with the coefficient optimized by training
2) Unstructured dense neural network



(a) Transient and steady-state cost



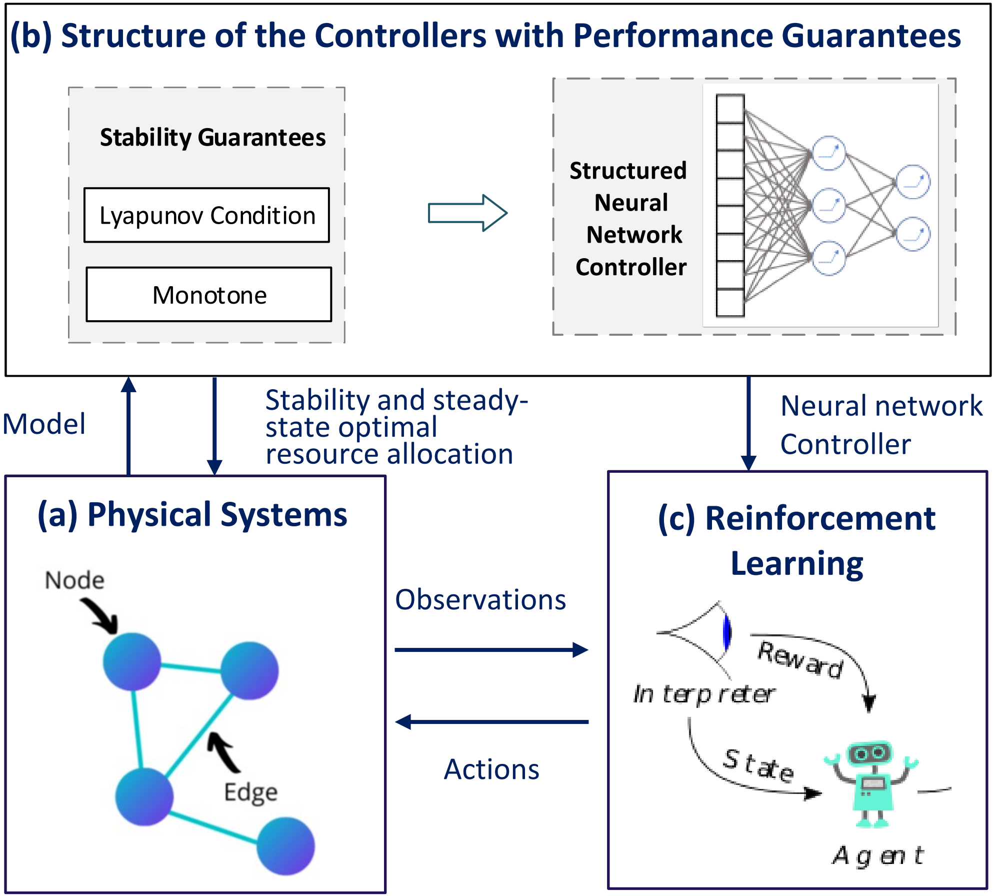
(b) Velocities under DenseNN



(c) Velocities under Neural-PI

5 Conclusions

- Derive structural properties of controllers satisfying performance guarantees
- Enforce the structures in the design of neural networks



Thank you!

- Feel free to contact me at wenqicui@uw.edu
- Online version and code of the above works can be found in

[1] Wenqi Cui, Yan Jiang, and Baosen Zhang. Reinforcement learning for optimal primary frequency control: A Lyapunov approach. IEEE Transactions on Power Systems, 2022

[2] Wenqi Cui, Yan Jiang, Baosen Zhang and Yuanyuan Shi. Structured Neural-PI Control for Networked Systems: Stability and Steady-State Optimality Guarantees. NeurIPS, 2023.