

A Proof

A.1 Proof of Theorem 1

We leverage the results in [49, 50] that the structure (5) with ReLU activation is a universal approximation for any convex function. However, ReLU activations are not strictly convex, and thus we design the Softplus- β activation. By showing that the structure (5) with Softplus- β activation can approximate neural networks with the ReLU activations arbitrarily closely when β is sufficiently large, we then prove that the structure (5) with Softplus- β can universally approximate any strictly convex function.

To prepare for the proof of Theorem 1, we first derive the following Lemma about the difference between ReLU activations and Softplus- β activation.

Lemma 3. *Consider the ReLU activation $\sigma_l^{\text{ReLU}}(x) := \max(x, 0)$. For all $x \in \mathbb{R}$ and $\Delta > 0$, we have $0 < \left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{ReLU}}(x) \right) < \Delta + \frac{1}{\beta} \log(2)$.*

Proof. Note that

$$\left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{ReLU}}(x) \right) = \left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{Softplus}\beta}(x) \right) + \left(\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) \right),$$

we prove the lemma by deriving the bound for $\left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{Softplus}\beta}(x) \right)$ and $\left(\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) \right)$ as follows

- (i) Since $\frac{d\sigma_l^{\text{Softplus}\beta}(x)}{dx} = e^{\beta x} / (1 + e^{\beta x}) \in (0, 1)$, then $0 < \left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{Softplus}\beta}(x) \right) < \Delta$ for $\Delta > 0$.
- (ii) Explicitly represent $\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x)$ yields

$$\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) = \begin{cases} \frac{1}{\beta} \log(1 + e^{\beta x}) - x, & x \geq 0 \\ \frac{1}{\beta} \log(1 + e^{\beta x}), & x < 0 \end{cases}$$

Thus,

$$\frac{d\left(\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) \right)}{dx} := \begin{cases} -1/(1 + e^{\beta x}), & x \geq 0 \\ e^{\beta x} / (1 + e^{\beta x}), & x < 0 \end{cases}$$

and therefore $\left(\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) \right) \leq \left(\sigma_l^{\text{Softplus}\beta}(0) - \sigma_l^{\text{ReLU}}(0) \right) = \frac{1}{\beta} \log(2)$. Note that $\frac{1}{\beta} \log(1 + e^{\beta x}) > \frac{1}{\beta} \log(e^{\beta x}) = x$, then $0 < \left(\sigma_l^{\text{Softplus}\beta}(x) - \sigma_l^{\text{ReLU}}(x) \right) \leq \frac{1}{\beta} \log(2)$.

Combining (i) and (ii), $0 < \left(\sigma_l^{\text{Softplus}\beta}(x + \Delta) - \sigma_l^{\text{ReLU}}(x) \right) < \Delta + \frac{1}{\beta} \log(2)$.

□

The proof of Theorem 1 is given below.

Proof. Previous works [49, 50] have shown that the structure (5) with ReLU activation is a universal approximation for any convex function. Hence, for any $q(\mathbf{z}) : \mathcal{Z} \mapsto \mathbb{R}$, there exists $g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}}$ constructed by (5) where the activation function is ReLU and satisfying $|g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}} - q(\mathbf{z})| < \frac{1}{2}\epsilon$. Note that

$$|g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - q(\mathbf{z})| \leq |g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}}| + |g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}} - q(\mathbf{z})|,$$

then it suffices to prove $|g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - q(\mathbf{z})| < \epsilon$ by showing the existence of $g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta}$ such that $|g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}}| < \frac{1}{2}\epsilon$ for all $\mathbf{z} \in \mathcal{Z}$.

Next, we compute the difference of the structure (5) with softplus- β and ReLU activations by tracing through the first layer to the last layer, under the same weights $\boldsymbol{\theta} = \left\{ W_{0:k-1}^{(z)}, W_{1:k-1}^{(o)}, b_{0:k-1} \right\}$.

The difference between the output of the first layer in $g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta}$ and $g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}}$ is

$$\mathbf{o}_1^{\text{Softplus}\beta} - \mathbf{o}_1^{\text{ReLU}} = \sigma_1^{\text{Softplus}\beta} \left(W_0^{(z)} \mathbf{z} + b_0 \right) - \sigma_1^{\text{ReLU}} \left(W_0^{(z)} \mathbf{z} + b_0 \right), \quad (10)$$

which yields $0 < \sigma_1^{\text{Softplus}\beta} - \sigma_1^{\text{ReLU}} \leq \frac{1}{\beta} \log(2) \mathbf{1}$ by Lemma 3.

The difference of the second layer is

$$\begin{aligned} \mathbf{o}_2^{\text{Softplus}\beta} - \mathbf{o}_2^{\text{ReLU}} &= \sigma_2^{\text{Softplus}\beta} \left(W_1^{(o)} \mathbf{o}_1^{\text{Softplus}\beta} + W_2^{(z)} \mathbf{z} + b_2 \right) - \sigma_2^{\text{ReLU}} \left(W_1^{(o)} \mathbf{o}_1^{\text{ReLU}} + W_2^{(z)} \mathbf{z} + b_2 \right) \\ &= \sigma_2^{\text{Softplus}\beta} \left(W_1^{(o)} \left(\mathbf{o}_1^{\text{Softplus}\beta} - \mathbf{o}_1^{\text{ReLU}} \right) + W_1^{(o)} \mathbf{o}_1^{\text{ReLU}} + W_2^{(z)} \mathbf{z} + b_2 \right) \\ &\quad - \sigma_2^{\text{ReLU}} \left(W_1^{(o)} \mathbf{o}_1^{\text{ReLU}} + W_2^{(z)} \mathbf{z} + b_2 \right). \end{aligned} \quad (11)$$

Since all the element of $W_1^{(o)}$ is positive, we have $0 < W_1^{(o)} \left(\mathbf{o}_1^{\text{Softplus}\beta} - \mathbf{o}_1^{\text{ReLU}} \right) \leq \frac{1}{\beta} \log(2) W_1^{(o)} \mathbf{1}$.

Applying Lemma 3 element-wise yields $0 \leq \sigma_2^{\text{Softplus}\beta} - \sigma_2^{\text{ReLU}} \leq \frac{1}{\beta} \log(2) W_1^{(o)} \mathbf{1} + \frac{1}{\beta} \log(2) \mathbf{1}$.

Similarly $0 \leq \sigma_3^{\text{Softplus}\beta} - \sigma_3^{\text{ReLU}} \leq \frac{1}{\beta} \log(2) W_2^{(o)} W_1^{(o)} \mathbf{1} + \frac{1}{\beta} \log(2) W_2^{(o)} \mathbf{1} + \frac{1}{\beta} \log(2) \mathbf{1}$. By induction

$$0 \leq \sigma_{l+1}^{\text{Softplus}\beta} - \sigma_{l+1}^{\text{ReLU}} \leq \frac{1}{\beta} \log(2) \left(\mathbf{1} + \left(\sum_{i=1}^l \prod_{j=i}^l W_j^{(o)} \mathbf{1} \right) \right) \quad (12)$$

Hence,

$$0 \leq g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}} \leq \frac{1}{\beta} \log(2) \left(1 + \left(\sum_{i=1}^{k-1} \prod_{j=i}^{k-1} W_j^{(o)} \mathbf{1} \right) \right), \quad (13)$$

where $\prod_{j=i}^{k-1} W_j^{(o)} := W_{k-1}^{(o)} W_{k-2}^{(o)} \cdots W_i^{(o)}$.

Let $\beta > \frac{2}{\epsilon} \log(2) \left(1 + \left(\sum_{i=1}^{k-1} \prod_{j=i}^{k-1} W_j^{(o)} \right) \mathbf{1} \right)$, then $0 \leq g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}} \leq \frac{1}{2} \epsilon$. We complete the proof using

$$|g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - q(\mathbf{z})| \leq |g(\mathbf{z}; \boldsymbol{\theta})^{\text{Softplus}\beta} - g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}}| + |g(\mathbf{z}; \boldsymbol{\theta})^{\text{ReLU}} - q(\mathbf{z})| < \frac{1}{2} \epsilon + \frac{1}{2} \epsilon = \epsilon. \quad \square$$

A.2 Proof of Theorem 2

Proof. At an equilibrium, the right side of (4b) equals to zero gives $\mathbf{y}^* = \bar{\mathbf{y}}$ and the corresponding set of equilibrium $\mathcal{E} = \{\mathbf{x}^*, \mathbf{s}^* | \mathbf{f}(\mathbf{x}^*, \mathbf{r}(\mathbf{s}^*)) = \mathbf{0}, \mathbf{y}^* = \bar{\mathbf{y}}, \mathbf{h}(\mathbf{x}^*) = \mathbf{y}^*\}$. We construct a Lyapunov function to prove that if there is a feasible equilibrium in \mathcal{E} , then the system is locally asymptotically stable around it.

We construct a Lyapunov function using the storage function in Assumption 2 and the Bregman distance in Lemma 2 as

$$V(\mathbf{x}, \mathbf{s})|_{\mathbf{x}^*, \mathbf{s}^*} = S(\mathbf{x}, \mathbf{x}^*) + B(\mathbf{s}, \mathbf{s}^*; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)}), \quad (14)$$

where the functions by construction satisfy $S(\mathbf{x}, \mathbf{x}^*) \geq 0$, $B(\mathbf{s}, \mathbf{s}^*; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)}) \geq 0$ with equality holds only when $\mathbf{x} = \mathbf{x}^*$ and $\mathbf{s} = \mathbf{s}^*$, respectively. Hence, $V(\mathbf{x}, \mathbf{s})|_{\mathbf{x}^*, \mathbf{s}^*}$ is a well-defined function that is positive definite and equals to zero at the equilibrium.

To prepare for the calculation of the time derivative of the Lyapunov function, we start by calculating the time derivative of the function $B(\mathbf{s}, \mathbf{s}^*; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)})$:

$$\begin{aligned} \dot{B}(\mathbf{s}, \mathbf{s}^*; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)}) &= \left(\nabla_{\mathbf{s}} g^{(I)}(\mathbf{s}; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)}) - \nabla_{\mathbf{s}} g^{(I)}(\mathbf{s}^*; \boldsymbol{\theta}^{(I)}, \boldsymbol{\beta}^{(I)}) \right)^\top \dot{\mathbf{s}} \\ &\stackrel{\textcircled{1}}{=} (\mathbf{r}(\mathbf{s}) - \mathbf{r}(\mathbf{s}^*))^\top (-\mathbf{y} - \mathbf{y}^*), \end{aligned} \quad (15)$$

where ① follows from $\nabla_s g^{(l)}(s; \theta^{(l)}, \beta^{(l)}) = \mathbf{r}(s)$ and $\dot{s} = -(\mathbf{y} - \bar{\mathbf{y}}) = -(\mathbf{y} - \mathbf{y}^*)$.

The time derivative of the Lyapunov function is

$$\begin{aligned} \dot{V}(\mathbf{x}, \mathbf{s})|_{\mathbf{x}^*, \mathbf{s}^*} &= \dot{S}(\mathbf{x}, \mathbf{x}^*) + \dot{B}(\mathbf{s}, \mathbf{s}^*; \theta, \beta) \\ &\stackrel{\textcircled{1}}{\leq} -\rho \|\mathbf{y} - \mathbf{y}^*\|^2 + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{u} - \mathbf{u}^*) + (\mathbf{r}(\mathbf{s}) - \mathbf{r}(\mathbf{s}^*))^\top (-\mathbf{y} - \mathbf{y}^*) \\ &\stackrel{\textcircled{2}}{=} -\rho \|\mathbf{y} - \mathbf{y}^*\|^2 + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{p}(-\mathbf{y} + \bar{\mathbf{y}}) - \mathbf{p}(-\mathbf{y}^* + \bar{\mathbf{y}})) \\ &\stackrel{\textcircled{3}}{\leq} -\rho \|\mathbf{y} - \mathbf{y}^*\|^2 \end{aligned} \quad (16)$$

where ① follows from the strict EIP property and equations derived in (15). The equality ② is derived by plugging in the controller design in (4a) where $\mathbf{u} = \mathbf{p}(-\mathbf{y} + \bar{\mathbf{y}}) + \mathbf{r}(s)$ and $\mathbf{u}^* = \mathbf{p}(-\mathbf{y}^* + \bar{\mathbf{y}}) + \mathbf{r}(s^*)$. The inequality ③ uses strictly monotone property of $\mathbf{p}(\cdot)$.

Therefore, $\dot{V}(\mathbf{x}, \mathbf{s})|_{\mathbf{x}^*, \mathbf{s}^*} \leq 0$ with equality only holds at the equilibrium. By Lyapunov stability theory in Proposition 1, the system is locally asymptotically stable around the equilibrium. \square

B Experiments

We demonstrate the effectiveness of the proposed neural-PI control in two dynamical systems: vehicle platooning and power system frequency control. All experiments are run with an NVIDIA Tesla T4 GPU with 16GB memory. For completeness, the figures highlighted in Section 6 are also shown below with more thorough discussions. Code is available at [this link](#).

B.1 Vehicle platooning

B.1.1 Problem statement

The first experiment is the vehicle platoon control in [3, 23] with m vehicles, where $\mathbf{u} \in \mathbb{R}^m$ is the control signal to adjust the velocities of vehicles and the output $\mathbf{y} \in \mathbb{R}^m$ is their actual velocities. The state is $\mathbf{x} = (\zeta, \mathbf{y})$, where $\zeta \in \mathbb{R}^m$ is the relative position of vehicles with $\zeta(0) \perp \text{Im}(\mathbf{1}_m)$ (namely, the vehicles are not in the same position at the time step 0). The dynamic model is

$$\begin{aligned} \dot{\zeta} &= \Gamma \mathbf{y}, \\ \dot{\mathbf{y}} &= \hat{\kappa} \left(-(\mathbf{y} - \boldsymbol{\lambda}^0) + \hat{\rho} \left(\mathbf{u} - \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \zeta \right) \right), \end{aligned} \quad (17)$$

where $\hat{\kappa} = \text{diag}(\kappa_1, \dots, \kappa_m)$, $\hat{\rho} = \text{diag}(\rho_1, \dots, \rho_m) \in \mathbb{R}^{m \times m}$ are constant diagonal matrices and $\kappa_i > 0, \rho_i > 0$ for all $i = 1, \dots, m$. The vector $\boldsymbol{\lambda}^0 = (\lambda_1^0, \dots, \lambda_m^0) \in \mathbb{R}^m$ reflects the default velocity of vehicles. The matrix $\mathbf{E} \in \mathbb{R}^{m \times e}$ is the incidence matrix that indicates the neighbouring relations for e pairs of neighbouring vehicles and satisfy $\ker(\mathbf{E}^\top) = \text{Im}(\mathbf{1}_m)$. The matrix $\Gamma := \mathbf{I}_m - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^\top$ extracts the relative velocities of vehicles by $\Gamma \mathbf{y}$. The diagonal matrix $\hat{\mathbf{D}} \in \mathbb{R}^{m \times m}$ represents the sensitivity to the relative distance of neighbouring vehicles.

B.1.2 Verification of Assumption 2

We start by verifying the uniqueness of \mathbf{x}^* for any $\mathbf{u}^* \in \mathbb{R}^m$. At the equilibrium, the right side of (17) equals zero gives

$$\left(-(\mathbf{y}^* - \boldsymbol{\lambda}^0) + \hat{\rho} \left(\mathbf{u}^* - \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \zeta^* \right) \right) = \mathbf{0}_m \text{ and } \Gamma \mathbf{y}^* = \mathbf{0}_m. \quad (18)$$

For a given $\mathbf{u}^* \in \mathbb{R}^m$, suppose there exists $\mathbf{x}_a^* = (\zeta_a^*, \mathbf{y}_a^*)$ and $\mathbf{x}_b^* = (\zeta_b^*, \mathbf{y}_b^*)$, $\mathbf{x}_a^* \neq \mathbf{x}_b^*$ such that (18) holds. Plugging in (18) gives

$$(\mathbf{y}_a^* - \mathbf{y}_b^*) + \hat{\rho} \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*) = \mathbf{0}_m \quad (19a)$$

$$\Gamma (\mathbf{y}_a^* - \mathbf{y}_b^*) = \mathbf{0}_m. \quad (19b)$$

Note that $\Gamma \mathbf{E} = \mathbf{E}$. Left multiplying (19a) with $(\mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*))^\top \Gamma$ yields $(\mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*))^\top \hat{\rho} \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*) = 0$, which holds if and only if $\mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*) = \mathbf{0}_m$ since $\hat{\rho} \succ 0$.

Hence, $(\mathbf{y}_a^* - \mathbf{y}_b^*) = -\hat{\rho} \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*) = \mathbf{0}_m$ and therefore $\mathbf{y}_a^* = \mathbf{y}_b^*$. Note that $\ker(\mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top) = \text{Im}(\mathbf{1}_m)$ and $\text{Im}(\Gamma) \perp \text{Im}(\mathbf{1}_m)$, thus $(\zeta_a^* - \zeta_b^*) \perp \text{Im}(\mathbf{1}_m)$. Hence, $\mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top (\zeta_a^* - \zeta_b^*) = \mathbf{0}_m$ if and only if $\zeta_a^* = \zeta_b^*$. Therefore, for every equilibrium $\mathbf{u}^* \in \mathbb{R}^m$, there is a unique $\mathbf{x}^* = (\zeta^*, \mathbf{y}^*) \in \mathbb{R}^n$ such that $\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}_n$.

Let the storage function be $S(\mathbf{x}, \mathbf{x}^*) = \frac{1}{2}(\mathbf{y} - \mathbf{y}^*)^\top \hat{\kappa}^{-1} \hat{\rho}^{-1}(\mathbf{y} - \mathbf{y}^*) + \frac{1}{2} \zeta^\top \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \zeta$. Then

$$\begin{aligned} \dot{S}(\mathbf{x}, \mathbf{x}^*) &= (\mathbf{y} - \mathbf{y}^*)^\top \hat{\rho}^{-1} \hat{\kappa}^{-1} \dot{\mathbf{y}} + \zeta^\top \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \dot{\zeta} \\ &= (\mathbf{y} - \mathbf{y}^*)^\top \hat{\rho}^{-1} \left(-(\mathbf{y} - \lambda^0) + \hat{\rho} (\mathbf{u} - \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \zeta) \right) + \zeta^\top \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \Gamma \mathbf{y} \\ &\stackrel{\textcircled{1}}{=} -\hat{\rho}^{-1} \|\mathbf{y} - \mathbf{y}^*\|_2^2 + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{u} - \mathbf{u}^*) \\ &\stackrel{\textcircled{2}}{\leq} -(\min_i \rho_i^{-1}) \|\mathbf{y} - \mathbf{y}^*\|_2^2 + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{u} - \mathbf{u}^*) \end{aligned}$$

where $\textcircled{1}$ follows from $(-(\mathbf{y}^* - \lambda^0) + \hat{\rho} (\mathbf{u}^* - \mathbf{E} \hat{\mathbf{D}} \mathbf{E}^\top \zeta^*)) = \mathbf{0}_m$ and $\mathbf{E}^\top \mathbf{y}^* = \mathbf{E}^\top \Gamma \mathbf{y}^* = \mathbf{0}_e$ by definition of equilibrium. The inequality $\textcircled{2}$ follows from $\hat{\rho} > 0$. Therefore, the dynamics in (17) satisfy conditions in Assumption 2.

B.1.3 Simulation and Visualization

Simulation and training setup We adopt the model setup and parameters in [3, 23]. The number of vehicles is $m = 20$. The sensitivity parameter is $\kappa_i = 1$ for all vehicles. The parameters λ_i^0 and ρ_i are randomly generated by $\lambda_i^0 \sim \text{uniform}[5, 6]$ and $\rho_i \sim \text{uniform}[1, 2]$, respectively. We train for 400 epochs, where each epoch trains with 300 trajectories with initial velocities $y_i(0) \sim \text{uniform}[5, 6]$ and initial integral variable $s_i(0) = 0$. The stepsize in time is set as $\Delta t = 0.02s$ and for $K = 300$ steps in a trajectory (Namely, each trajectory evolves 6s).

We implement control law in \mathbf{u} to realize a specific output agreement at $\bar{\mathbf{y}}$ and reduce the transient cost. The transient cost is set to be $J(\mathbf{y}, \mathbf{u}) = \sum_{k=1}^K \|\mathbf{y}(k\Delta t) - \bar{\mathbf{y}}\|_1 + \hat{c} \|\mathbf{u}(k\Delta t)\|_2^2$, where $\hat{c} = \text{diag}(c_1, \dots, c_m)$ with $c_i \sim \text{uniform}[0.025, 0.075]$. The loss function in training is set to be the same as $J(\mathbf{y}, \mathbf{u})$, such that neural networks are optimized to reduce transient cost through training. The neural PI controller can be trained by most model-based or model-free algorithms, and we use the model-based framework in [8, 53] by embedding the system dynamic model in the computation graph shown in Figure 4 and training Neural-PI by gradient descent through $J(\mathbf{y}, \mathbf{u})$.

Controller performances. We compare the performance of 1) Neural-PI: the learned structured Neural-PI controllers parametrized by (7) with three layers and 20 neurons in each hidden layer. The neural networks are updated using Adam with learning rate initializes at 0.05 and decays every 50 steps with a base of 0.7. 2) DenseNN-PI: The proportional and integral terms are parameterized by dense neural networks (5) with three layers, 20 neurons in each hidden layer, and unconstrained weights. The neural networks are updated using Adam with learning rate initializes at 0.035 and decays every 50 steps with a base of 0.7. 3) Linear-PI: linear PI control where $\mathbf{p}(\bar{\mathbf{y}} - \mathbf{y}) := \mathbf{K}_P(\bar{\mathbf{y}} - \mathbf{y})$, $\mathbf{r}(s) := \mathbf{K}_I(s)$ with \mathbf{K}_P and \mathbf{K}_I being the trainable proportional and integral coefficients. The coefficients are updated using Adam with learning rate initializes at 0.03 and decays every 50 steps with a base of 0.7. All of them have no communication constraints. All of the controllers are trained using 5 random seeds. The training time is shown in Table 1.

Table 1: Training time for vehicle platoon control

Method	Average Training time (s)	Standard Deviation (s)
Neural-PI	5232.36	30.55
DenseNN-PI	3567.01	16.28
Linear-PI	1836.93	10.09

The average batch loss during epochs of training with 5 seeds is shown in Figure 7(a). All of the three methods converge, with the Neural-PI achieves the lowest cost. Figure 7(b) shows the transient

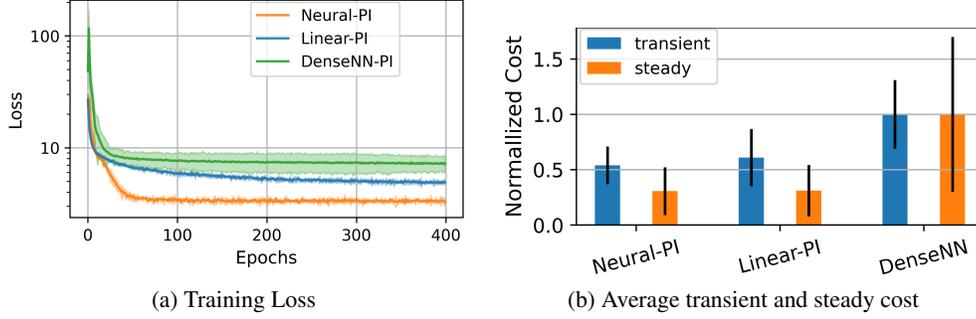


Figure 7: (a) The average batch loss during epochs of training with 5 seeds. All converge, with the Neural-PI achieving the lowest cost. (b) The average transient cost and steady-state cost with error bar on 100 testing trajectories starting from randomly generated initial states. Neural-PI achieves a transient cost that is much lower than others. DenseNN without structured design has both high costs in transient and steady-state performances.

and steady-state cost on 100 testing trajectories starting from randomly generated initial states. The steady-state cost is $C(\mathbf{y}, \mathbf{u}) = \|\mathbf{y}(15) - \bar{\mathbf{y}}\|_1 + \hat{c}\|\mathbf{u}(15)\|_2^2$, where we use the variables at the time $t = 15s$ since the dynamics approximately enter the steady state after $t = 15s$ as we will show later in simulation. Neural-PI and Linear-PI have the lowest steady-state cost, and the output reaches $\bar{\mathbf{y}}$ as guaranteed by Theorem 2. Neural-PI also achieves a transient cost that is much lower than others. By contrast, DenseNN-PI without structured design has both high costs in transient and steady-state performances.

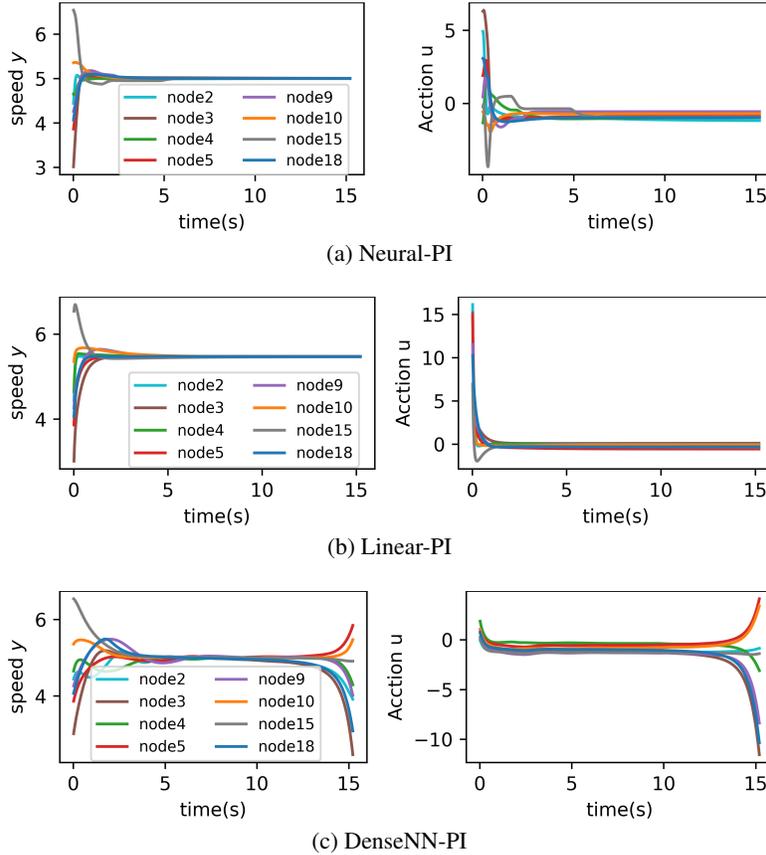


Figure 8: Dynamics of velocity \mathbf{y} and control action \mathbf{u} with $\bar{\mathbf{y}} = 5\text{m/s}$. (a) Neural-PI stabilizes to $\bar{\mathbf{y}}$ quickly. (b) Linear-PI achieves output tracking with high control effort. (c) DenseNN leads to unstable behavior.

Given $\bar{y} = 5\text{m/s}$, Figure 8 shows the dynamics of velocity \mathbf{y} and control action \mathbf{u} on 8 nodes under the three methods. As guaranteed by Theorem 2, Neural-PI in Figure 8(a) and Linear-PI in Figure 8(b) reaches the required speed $\bar{y} = 5\text{m/s}$. However, Linear-PI has slower convergence and much larger control efforts compared with Neural-PI. Even though DenseNN-PI achieves finite loss both in training and testing, Figure 8(c) actually exhibits unstable behaviors. In particular, DenseNN-PI appears to be stable until about 10s, but states blows up quickly after that. Therefore, enforcing stabilizing structures is essential.

B.2 Power systems frequency control

B.2.1 Problem statement

The second experiment is the power system frequency control on the IEEE 39-bus New England system [52] shown in Figure 9, where $\mathbf{u} \in \mathbb{R}^m$ is the control signal to adjust the power injection from generators and the output $\mathbf{y} \in \mathbb{R}^m$ is the rotating speed (i.e., frequency) of generators. The objective is to stabilize generators at the required frequency $\bar{y} = 60\text{Hz}$ at the steady state while minimizing the transient control cost. The state is $\mathbf{x} = (\boldsymbol{\delta}, \mathbf{y})$, where $\boldsymbol{\delta} \in \mathbb{R}^m$ is the rotating angle of generators in the center-of-inertia coordinates with $\boldsymbol{\delta}(0) \perp \text{Im}(\mathbf{1}_m)$ [54]. The model of power systems reflects the transmission of electricity from generators to loads through power transmission lines and is represented as follows:

$$\begin{aligned} \dot{\boldsymbol{\delta}} &= \boldsymbol{\Gamma}\mathbf{y}, \\ \hat{\mathbf{M}}\dot{\mathbf{y}} &= -\hat{\mathbf{D}}(\mathbf{y} - \bar{\mathbf{y}}) - \mathbf{d} + \mathbf{u} - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top\boldsymbol{\delta}), \end{aligned} \quad (20)$$

where $\hat{\mathbf{M}} = \text{diag}(M_1, \dots, M_m)$, $\hat{\mathbf{D}} = \text{diag}(D_1, \dots, D_m)$ with $M_j > 0$ and $D_j > 0$ being the inertia and damping constant of generator j , respectively. The vector \mathbf{d} is the net load of the system. The matrix $\mathbf{E} \in \mathbb{R}^{m \times e}$ is the incidence matrix corresponding to the topology of the power network with e transmission lines and satisfying $\ker(\mathbf{E}^\top) = \text{Im}(\mathbf{1}_m)$. The matrix $\boldsymbol{\Gamma} := \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top$ extracts the relative rotating speed of generators by $\boldsymbol{\Gamma}\mathbf{y}$. The diagonal matrix $\hat{\mathbf{b}} = \text{diag}(b_1, \dots, b_e) \in \mathbb{R}^{e \times e}$ with $b_j > 0$ being the susceptance of the j -th transmission line.

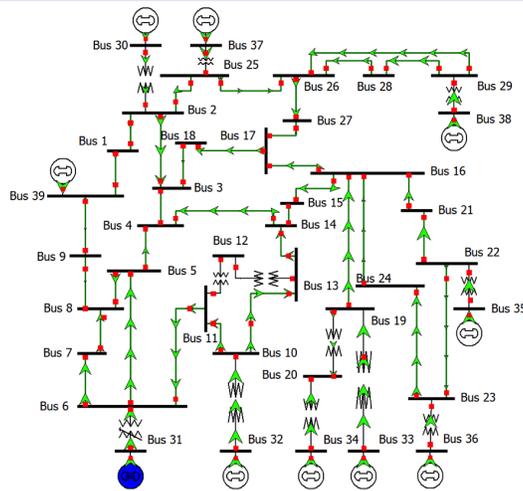


Figure 9: IEEE 39-bus test system [52]

We adopt a common assumption in literature that the power system operates with $\boldsymbol{\delta}$ satisfying $\mathcal{H} = \{\boldsymbol{\delta} \mid [\mathbf{E}^\top\boldsymbol{\delta}]_j \in (-\pi/2, \pi/2) \forall j = 1, \dots, e\}$, where $[\mathbf{E}^\top\boldsymbol{\delta}]_j$ is the angle difference between the generators in head and tail of the j -th transmission line [55–57]. This range is sufficiently large to include almost all practical scenarios [55–57].

B.2.2 Verification of Assumption 2

At the equilibrium, the right side of (17) equals zero gives

$$-\hat{D}(\mathbf{y}^* - \bar{\mathbf{y}}) - \mathbf{d} + \mathbf{u}^* - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*) = \mathbf{0}_m \text{ and } \boldsymbol{\Gamma}\mathbf{y}^* = \mathbf{0}_m. \quad (21)$$

We start by verifying the uniqueness of \mathbf{x}^* for any $\mathbf{u}^* \in \mathcal{U}$ where (21) has a feasible solution such that $\boldsymbol{\delta} \in \mathcal{H}$. For a given $\mathbf{u}^* \in \mathcal{U}$, suppose there exists $\mathbf{x}_a^* = (\boldsymbol{\delta}_a^*, \mathbf{y}_a^*)$ and $\mathbf{x}_b^* = (\boldsymbol{\delta}_b^*, \mathbf{y}_b^*)$, $\mathbf{x}_a^* \neq \mathbf{x}_b^*$ such that (21) holds. Plugging in (21) gives

$$\hat{D}(\mathbf{y}_a^* - \mathbf{y}_b^*) + \mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*)) = \mathbf{0}_m \quad (22a)$$

$$\boldsymbol{\Gamma}(\mathbf{y}_a^* - \mathbf{y}_b^*) = \mathbf{0}_m. \quad (22b)$$

Note that $\boldsymbol{\Gamma}\mathbf{E} = \mathbf{E}$. Left multiplying (22a) with $(\mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*)))^\top \boldsymbol{\Gamma}\hat{D}^{-1}$ yields $(\mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*)))^\top \hat{D}^{-1}(\mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*))) = 0$, which holds if and only if $(\mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*))) = \mathbf{0}_m$ since $\hat{D}^{-1} \succ 0$. Plugging in (22a) gives $\hat{D}(\mathbf{y}_a^* - \mathbf{y}_b^*) = \mathbf{0}_m$, which holds if and only if $\mathbf{y}_a^* = \mathbf{y}_b^*$ since $D_i > 0$ for all $i = 1, \dots, m$.

Left multiplying $(\mathbf{E}\hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*))) = \mathbf{0}_m$ with $(\boldsymbol{\delta}_a^* - \boldsymbol{\delta}_b^*)^\top$ yields

$$\begin{aligned} 0 &= (\mathbf{E}^\top \boldsymbol{\delta}_a^* - \mathbf{E}^\top \boldsymbol{\delta}_b^*)^\top \hat{\mathbf{b}}(\sin(\mathbf{E}^\top \boldsymbol{\delta}_a^*) - \sin(\mathbf{E}^\top \boldsymbol{\delta}_b^*)) \\ &= \sum_{j=1}^e b_j ([\mathbf{E}^\top \boldsymbol{\delta}_a^*]_j - [\mathbf{E}^\top \boldsymbol{\delta}_b^*]_j) (\sin([\mathbf{E}^\top \boldsymbol{\delta}_a^*]_j) - \sin([\mathbf{E}^\top \boldsymbol{\delta}_b^*]_j)). \end{aligned} \quad (23)$$

Since $b_j > 0$ and $\sin(\cdot)$ is strictly increasing in $(-\pi/2, \pi/2)$, (23) holds if and only if $\mathbf{E}^\top(\boldsymbol{\delta}_a^* - \boldsymbol{\delta}_b^*) = \mathbf{0}_e$. Note that $\text{Im}(\boldsymbol{\Gamma}) \perp \text{Im}(\mathbf{1}_m)$, thus $(\boldsymbol{\delta}_a^* - \boldsymbol{\delta}_b^*) \perp \text{Im}(\mathbf{1}_m)$. Hence, (23) holds if and only if $\boldsymbol{\delta}_a^* = \boldsymbol{\delta}_b^*$. Therefore, for every equilibrium $\mathbf{u}^* \in \mathcal{U}$, there is a unique $\mathbf{x}^* = (\boldsymbol{\delta}^*, \mathbf{y}^*) \in \mathbb{R}^n$ such that $\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}_n$.

Let the storage function be $S(\mathbf{x}, \mathbf{x}^*) = \frac{1}{2}(\mathbf{y} - \mathbf{y}^*)^\top \hat{\mathbf{M}}(\mathbf{y} - \mathbf{y}^*) - \mathbf{1}_e^\top \hat{\mathbf{b}}(\cos(\mathbf{E}^\top \boldsymbol{\delta}) - \cos(\mathbf{E}^\top \boldsymbol{\delta}^*)) - (\mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*))^\top (\boldsymbol{\delta} - \boldsymbol{\delta}^*)$. Note that $-\mathbf{1}_e^\top \hat{\mathbf{b}}(\cos(\mathbf{E}^\top \boldsymbol{\delta}))$ is strictly convex in \mathcal{H} , thus the Bregman distance $-\mathbf{1}_e^\top \hat{\mathbf{b}}(\cos(\mathbf{E}^\top \boldsymbol{\delta}) - \cos(\mathbf{E}^\top \boldsymbol{\delta}^*)) - (\mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*))^\top (\boldsymbol{\delta} - \boldsymbol{\delta}^*) \geq 0$ with equality holds only when $\boldsymbol{\delta} = \boldsymbol{\delta}^*$.

The time derivative is

$$\begin{aligned} \dot{S}(\mathbf{x}, \mathbf{x}^*) &= (\mathbf{y} - \mathbf{y}^*)^\top \hat{\mathbf{M}}\dot{\mathbf{y}} + (\mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}) - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*))^\top \dot{\boldsymbol{\delta}} \\ &= (\mathbf{y} - \mathbf{y}^*)^\top (-\hat{D}(\mathbf{y} - \bar{\mathbf{y}}) - \mathbf{d} + \mathbf{u} - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta})) \\ &\quad + (\mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}) - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*))^\top \boldsymbol{\Gamma}\mathbf{y} \\ &\quad - (\mathbf{y} - \mathbf{y}^*)^\top \underbrace{(-\hat{D}(\mathbf{y}^* - \bar{\mathbf{y}}) - \mathbf{d} + \mathbf{u}^* - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*))}_{=\mathbf{0}_m} \\ &\stackrel{\textcircled{1}}{=} -(\mathbf{y} - \mathbf{y}^*)^\top \hat{D}(\mathbf{y} - \mathbf{y}^*) + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{u} - \mathbf{u}^*) \\ &\quad - (\mathbf{y}^*)^\top (\mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}) - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*)) \\ &\stackrel{\textcircled{2}}{\leq} -(\min_i D_i) \|\mathbf{y} - \mathbf{y}^*\|_2^2 + (\mathbf{y} - \mathbf{y}^*)^\top (\mathbf{u} - \mathbf{u}^*) \end{aligned}$$

where ① follows from $(-\hat{D}(\mathbf{y}^* - \bar{\mathbf{y}}) - \mathbf{d} + \mathbf{u}^* - \mathbf{E}\hat{\mathbf{b}}\sin(\mathbf{E}^\top \boldsymbol{\delta}^*)) = \mathbf{0}_m$ by definition of equilibrium. The relation ② follows from $\mathbf{E}^\top \mathbf{y}^* = \mathbf{E}^\top \boldsymbol{\Gamma}\mathbf{y}^* = \mathbf{0}_e$ and $D_i > 0$ for all $i = 1, \dots, m$. Therefore, the dynamics (20) of the power system frequency control satisfies conditions in Assumption 2.

B.2.3 Simulation and Visualization

Simulation Setup We conduct experiments on the IEEE New England 10-machine 39-bus (NE39) power network with parameters given in [52, 8]. We implement control law for power output \mathbf{u} of generators to realize the track of frequency at 60Hz and reduce the power generation cost. The state $\boldsymbol{\delta}$ is initialized as the solution of power flow at the nominal frequency and \mathbf{s} is initialized as 0.

The number of epochs and batch size are 400 and 300, respectively. The step-size in time is set as $\Delta t = 0.01s$ and the number of time stages in a trajectory for training is $K = 400$.

Apart from the accumulated frequency deviation, an important metric for the frequency control problem is the maximum frequency deviation (also known as the frequency nadir) after a disturbance [8]. Hence, the transient cost is set to be $J(\mathbf{y}, \mathbf{u}) = \sum_{i=1}^n (\max_{k=1, \dots, K} |y_i(k\Delta t) - \bar{y}| + 0.05 \sum_{k=1}^K |y_i(k\Delta t) - \bar{y}| + 0.005 \sum_{k=1}^K (u_i(k\Delta t))^2)$. The loss function in training is $J(\mathbf{y}, \mathbf{u})$, such that neural networks are optimized to reduce transient cost. The neural PI controller can be trained by most model-based or model-free algorithms, and we use the model-based framework in [8, 53] by embedding the system dynamic model in the computation graph shown in Figure 4 and training Neural-PI by gradient descent through $J(\mathbf{y}, \mathbf{u})$.

Two major goals of this experiment is

- 1) *Verifies the robustness of the controller under parameter changes.* Note that the load \mathbf{d} is a parameter in the dynamics (20). In particular, power system operator emphasizes on the ability of the system to withstand a big disturbance such as a step load change. To this end, we train and test controllers by randomly picking at most three generators to have a step load change uniformly distributed in $\text{uniform}[-1, 1]$ p.u., where 1p.u.=100 MW is the base unit of power for the IEEE-NE39 test system.
- 2) *Verifies the performances under communication constraints.* Most systems do not have fully connected real-time communication capabilities, so the controller needs to respect the communication constraints and we show the flexibility of Neural-PI control under different communication structures.

Controller Performances. We compare the performance of Neural-PI controller where 1) all the nodes can communicate 2) half of the nodes can communicate and 3) none of the nodes can communicate (thus the controller is decentralized), respectively. All neural-PI controllers are parameterized by (7) and (8) where each SCNN has three layers and 20 neurons in each hidden layer. The neural networks are updated using Adam with the learning rate initializes at 0.05 and decays every 50 steps with a base of 0.7. We compare against the following two benchmarks where all the nodes can communicate: 4) DenseNN-PI-Full: Dense neural networks (5) with three layers, 20 neurons in each hidden layer, and unconstrained weights. The neural networks are updated using Adam with a learning rate initializes at 0.01 and decays every 50 steps with a base of 0.7. Note that DenseNN needs such a small learning rate to let the training converge, the reason is that DenseNN may lead to unstable behaviors that we will see later. 5) Linear-PI-Full: linear PI control where $\mathbf{p}(\bar{\mathbf{y}} - \mathbf{y}) := \mathbf{K}_P(\bar{\mathbf{y}} - \mathbf{y})$, $\mathbf{r}(s) := \mathbf{K}_I(s)$ with \mathbf{K}_P and \mathbf{K}_I being the trainable proportional and integral coefficients. The coefficients are updated using Adam with the learning rate initializes at 0.08 and decays every 50 steps with a base of 0.7. All of the controllers are trained using 5 random seeds. The training time is shown in Table 2.

Table 2: Training time for power system frequency control

Method	Average Training time (s)	Standard Deviation (s)
Neural-PI-Full	4373.52	64.58
Neural-PI-Half	8034.92	115.26
Neural-PI-Dec	23549.34	300.95
DenseNN-PI-Full	2193.84	21.22
Linear-PI-Full	981.65	11.19

The average batch loss during epochs of training with 5 seeds is shown in Figure 10(a). All converge, with the Neural-PI-Full achieving the lowest cost. Figure 10(b) shows the average transient cost and steady-state cost with error bar on 100 testing trajectories subject to random step load changes. The steady-state cost is $C(\mathbf{y}, \mathbf{u}) = 0.05\|\mathbf{y}(15) - \bar{\mathbf{y}}\|_1 + 0.005\|\mathbf{u}(15)\|_2^2$, where we use the variables at the time $t = 15s$ since the dynamics approximately enter the steady state after $t = 15s$ as we will show later in simulation. Neural-PI-Full achieves the lowest transient and steady-state cost. Notably, the steady-state cost significantly decreases with increased communication capability. The reason is that communication serves to better allocated control efforts such that they can maintain output

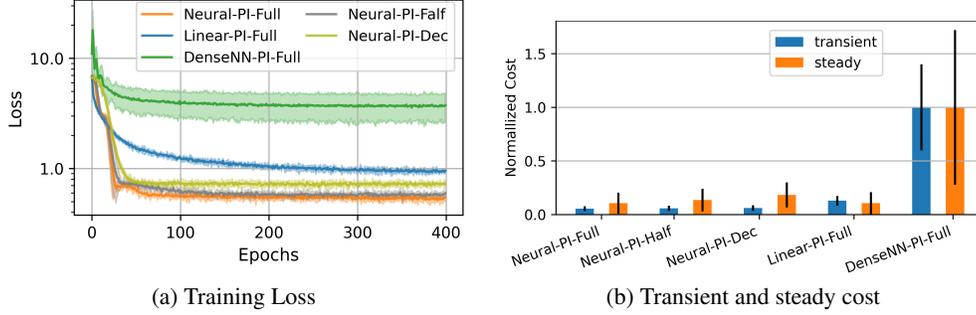


Figure 10: (a) Average batch loss during epochs of training with 5 seeds. All converge, with the Neural-PI achieving the lowest cost. (b) The average transient cost and steady-state cost with error bar on 100 testing trajectories subject to random step load changes. Neural-PI achieves a transient cost that is much lower than others. The steady-state cost significantly decreases with increased communication capability. DenseNN without structured design has both high costs in transient and steady-state performances.

Table 3: The average transient cost on 100 testing trajectories starting from randomly generated initial states

Number of training trajectories	Neural-PI	Linear-PI	DenseNN
5	0.1328	0.1915	1.0
10	0.1300	0.1865	0.9833
50	0.1257	0.1838	0.9624
100	0.1234	0.1816	0.9214
300	0.1233	0.1815	0.5347

tracking with smaller control costs. Again, DenseNN without structured design has high costs both in transient and in steady state.

With a step load change at 0.5s, Figure 11 shows the dynamics of frequency y and control action u on 7 nodes under the five methods. Again, DenseNN-PI-Full in Figure 11(e) exhibits unstable behavior with large oscillations. As guaranteed by Theorem 2, Neural-PI in Figure 11(a-c) reaches the required frequency $\bar{y} = 60\text{Hz}$, but the speed of convergence is lower for reduced communication capabilities. Hence, the guarantees provided by the structured Neural-PI controllers are robust to parameter changes and communication constraints, which have significant practical importance.

Performance with different numbers of training trajectories. Table 3 compares the transient cost attained by different controllers trained with different numbers of trajectories. For both Linear-PI and Neural-PI, training with 5 trajectories for each epoch has already achieved a similar cost as training with 300 trajectories. By contrast, unstructured DenseNN requires a much larger amount of training data to reduce transient costs on testing trajectories. Therefore, the stabilizing structure significantly reduces the requirement for the number of samples to learn well.

The impact of disturbances and noises. The satisfaction of the Lyapunov condition is robust to disturbances in the system parameters and does not need to know how large the disturbances are, as shown in the proof Theorem 2 and Remark 2. Therefore, if there is a sudden change in the load levels, the proposed controller design still stabilizes the system and tracks the required frequency at 60Hz. In Figure 12(a), we demonstrate the system dynamics after two disturbances in load. In Figure 12(b)-(c), we add noises in both data measurement and dynamics with the signal-to-noise ratio being 5 dB (much larger than typical measurement noises). The results show that the systems are input-to-state stable, i.e., that bounded noise will lead to bounded states. Incorporating noise in rigorous theoretical analysis is an important future direction for us.

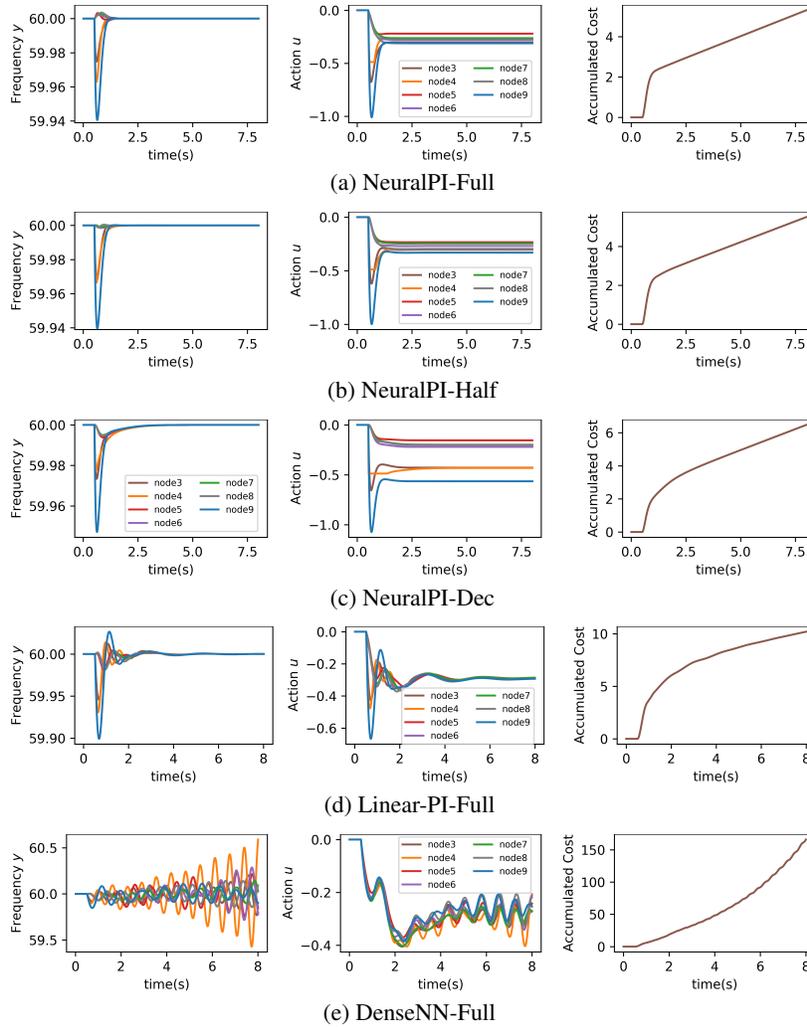


Figure 11: Dynamics of frequency y , control action u and accumulated cost on 7 nodes with $\bar{y} = 60\text{Hz}$ and a step load change at 0.5s. (a) Neural-PI when all nodes can communicate (b) Neural-PI when half of nodes can communicate, (c) Neural-PI when none nodes can communicate. The control with different communication capability all stabilize the system to the required $\bar{y} = 60\text{Hz}$. (d) Linear-PI-Full is stable but has slower convergence. (e) DenseNN-PI-Full leads to large frequency deviations and oscillations.

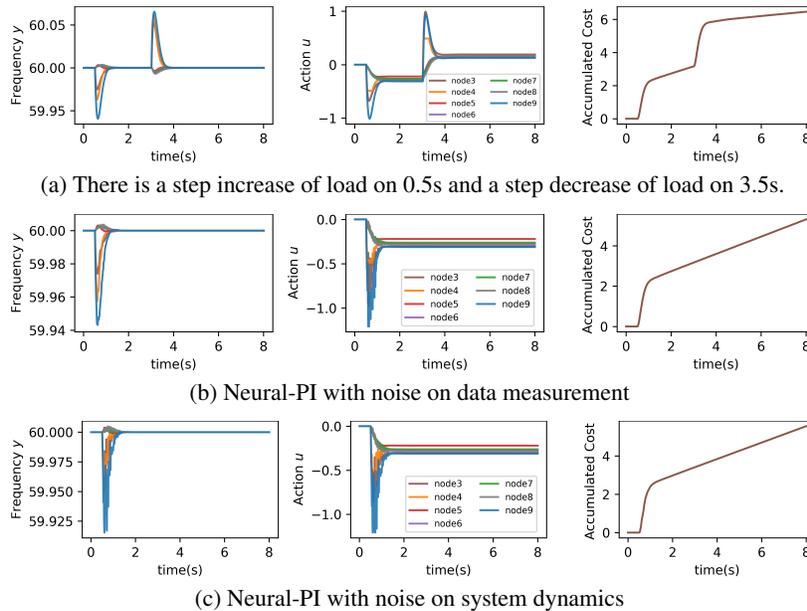


Figure 12: Frequency restoration to 60Hz after the disturbances/noises are all maintained.