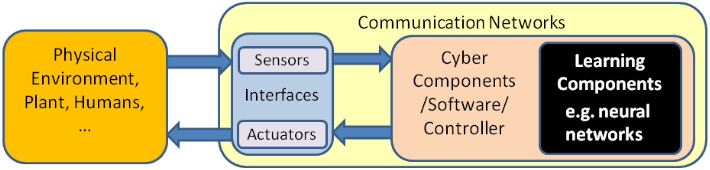
Reachability Analysis and Safety Verification for Neural Network Control Systems

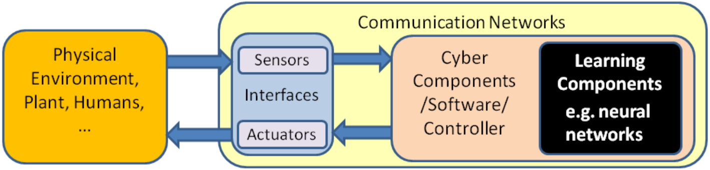
Weiming Xiang, Xiaodong Yang, Hoang-Dung Tran, Taylor T. Johnson

Department of Electrical Engineering and Computer Science
Institute for Software Integrated Systems
Vanderbilt University

Learning Enabled CPS



Learning Enabled CPS



- Plenty of learning techniques available due to the development of machine learning.
- High control performance, adaptive to the changes of environment, model-free design methodology, etc.
- Especially suitable for complex CPS and complicated control tasks, e.g., nonlinear, high-dimensional, time-varying, model unavailable.
- Learning components treated as **black boxes: non-transparent, incomplete, unpredictable, adversarial examples, etc**.

Safety for Learning Enabled CPS

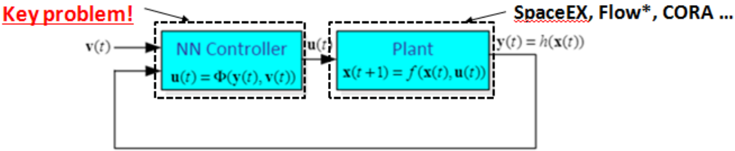


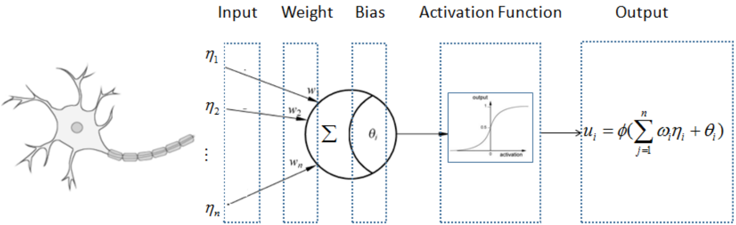
Figure: Illustration of neural network control systems

- Neural network control systems (closed-loop systems).
- Reachability of plant (using existing reachability tools).
- Reachability of neural network controller (**Key Problem!**)

Neuron

$$u_i = \phi\left(\sum_{j=1}^{n} \omega_{ij}\eta_j + \theta_i\right) \quad (1)$$

where η_j is the jth input of the ith neuron, ω_{ij} is the weight from the jth input to the ith neuron, θ_i is called the bias of the ith neuron, u_i is the output of the ith neuron, $\phi(\cdot)$ is the activation function.



Neural Network

$$\mathbf{u}^{[L]} = \Phi(\boldsymbol{\eta}^{[0]}) \tag{2}$$

where $\Phi(\cdot) \triangleq \phi_L \circ \phi_{L-1} \circ \cdots \circ \phi_1(\cdot)$

- **Neural Network**

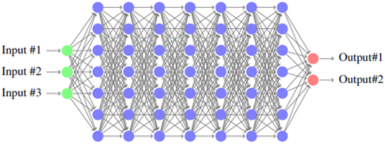


Figure: Illustration of neural networks

Neural Network Control Systems

Discrete-Time Systems

- System:

$$\begin{cases} \mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = h(\mathbf{x}(t)) \end{cases} \tag{3}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the system state, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ and $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ are control input and controlled output.

- Neural network controller: $\mathbf{u}(t) = \Phi(\mathbf{y}(t), \mathbf{v}(t))$.
- Closed-loop:

$$\begin{cases} \mathbf{x}(t+1) = f(\mathbf{x}(t), \Phi(\eta(t))) \\ \mathbf{y}(t) = h(\mathbf{x}(t)) \end{cases} \tag{4}$$

where $\eta(t) = [\mathbf{y}^\top(t)\ \mathbf{v}^\top(t)]^\top$.

Neural Network Control Systems

Continuous-Time Sampled Systems

- System:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = h(\mathbf{x}(t)) \end{cases} . \tag{5}$$

- Sampled neural network controller: $\mathbf{u}(t) = \Phi(\boldsymbol{\eta}(t_k)),\ t \in [t_k, t_{k+1})$.
- Closed-loop:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \Phi(\boldsymbol{\eta}(t_k))) \\ \mathbf{y}(t) = h(\mathrm{x}(t)) \end{cases} ,\ t \in [t_k, t_{k+1}) \tag{6}$$

where $\boldsymbol{\eta}(t_k) = [\mathbf{y}^\top(t_k)\ \mathbf{v}^\top(t_k)]^\top$.

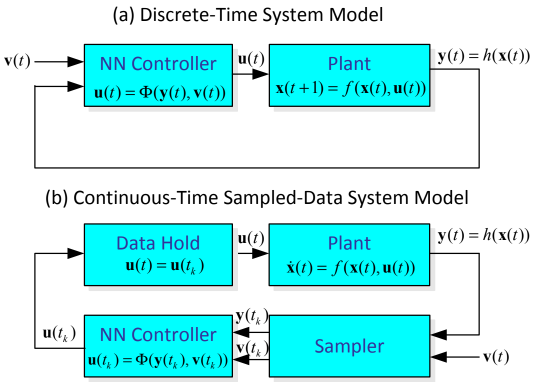


Figure: Illustration for closed-loop systems with neural network controllers

Problem Formulation

Output Set

Given a feedforward neural netowrk in the form of (2) and an input set $\mathcal{H}$, the following set

$$\mathcal{U} = \left\{ \mathbf{u}^{[L]} \in \mathbb{R}^{n_u} \mid \mathbf{u}^{[L]} = \Phi(\boldsymbol{\eta}^{[0]}),\ \boldsymbol{\eta}^{[0]} \in \mathcal{H} \right\} \tag{7}$$

is called the output set of feedforward neural netowrk (2).

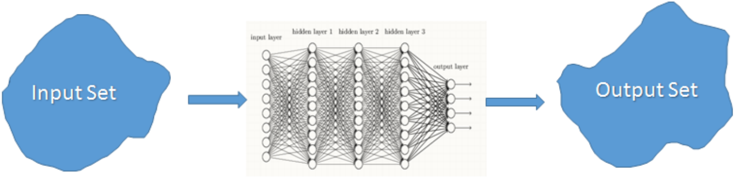


Figure: Illustration of output set

Problem Formulation

Reachable Set Estimation of Feedforward Neural Network

Given a bounded input set $\mathcal{H}$ and a feedforward neural netowrk described by (2), how does one find a set $\mathcal{U}_e$ such that $\mathcal{U} \subseteq \mathcal{U}_e$, and make the estimation set $\mathcal{U}_e$ **as small as possible**?

Problem Formulation

Reachable Set Estimation of Feedforward Neural Network

Given a bounded input set $\mathcal{H}$ and a feedforward neural netowrk described by (2), how does one find a set $\mathcal{U}_e$ such that $\mathcal{U} \subseteq \mathcal{U}_e$, and make the estimation set $\mathcal{U}_e$ **as small as possible**?

Reachable Set of Neural Network Control Systems

Given a neural network control system in the form of (4) with initial set $\mathcal{X}_0$ and input set $\mathcal{V}$, the reachable set at time t is

$$\mathcal{R}(t) = \{\mathbf{x}(t; \mathbf{x}_0, \mathbf{v}(\cdot)) \in \mathbb{R}^{n_x} \mid \mathbf{x}_0 \in \mathcal{X}_0, \mathbf{v}(t) \in \mathcal{V}\} \tag{8}$$

and the reachable set over time interval $[t_0, t_f]$ is defined by

$$\mathcal{R}([t_0, t_f]) = \bigcup\nolimits_{t \in [t_0, t_f]} \mathcal{R}(t). \tag{9}$$

Problem Formulation

Reachable Set Estimation of Neural Network Control Systems

Given a time t, how does one find the set $\mathcal{R}_e(t)$ such that $\mathcal{R}(t) \subseteq \mathcal{R}_e(t)$, given a bounded initial set $\mathcal{X}_0$ and an input set $\mathcal{V}$?

Safety of Neural Network Control Systems

Safety specification $\mathcal{S}$ formalizes the safety requirements for state $\mathbf{x}(t)$ of neural network control system (4). The neural network control system (4) is safe over time interval $[t_0, t_f]$ if and only if the following condition is satisfied:

$$\mathcal{R}([t_0, t_f]) \cap \neg\mathcal{S} = \emptyset \tag{10}$$

where $\neg$ is the symbol for logical negation.

Problem Formulation

Safety Verification of Neural Network Control Systems

How does one verify the safety requirement described by (10), given a neural network control system (4) with a bounded initial set $\mathcal{X}_0$ and an input set $\mathcal{V}$ and a safety specification $\mathcal{S}$?

Problem Formulation

Safety Verification of Neural Network Control Systems

How does one verify the safety requirement described by (10), given a neural network control system (4) with a bounded initial set $\mathcal{X}_0$ and an input set $\mathcal{V}$ and a safety specification $\mathcal{S}$?

Lemma (over-approximation is sufficient for verification)

Consider a neural network control system in the form of (4) and a safety specification $\mathcal{S}$, the neural network control system is safe in time interval $[t_0, t_f]$ if the following condition is satisfied

$$\mathcal{R}_e([t_0, t_f]) \cap \neg\mathcal{S} = \emptyset \tag{11}$$

where $\mathcal{R}([t_0, t_f]) \subseteq \mathcal{R}_e([t_0, t_f])$.

Problem Formulation

Safety Verification of Neural Network Control Systems

How does one verify the safety requirement described by (10), given a neural network control system (4) with a bounded initial set $\mathcal{X}_0$ and an input set $\mathcal{V}$ and a safety specification $\mathcal{S}$?

Lemma (over-approximation is sufficient for verification)

Consider a neural network control system in the form of (4) and a safety specification $\mathcal{S}$, the neural network control system is safe in time interval $[t_0, t_f]$ if the following condition is satisfied

$$\mathcal{R}_e([t_0, t_f]) \cap \neg\mathcal{S} = \emptyset \tag{11}$$

where $\mathcal{R}([t_0, t_f]) \subseteq \mathcal{R}_e([t_0, t_f])$.

Key Problem: How to find a tight over-approximation?

Hyper-Rectangular Sets

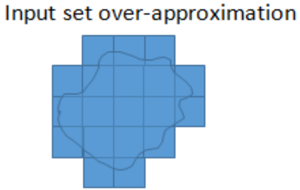
$$\mathcal{P}_i = \mathcal{I}_{1,m_1} \times \cdots \times \mathcal{I}_{n,m_n}, \ i \in \{1, 2, \ldots, \prod\nolimits_{s=1}^{n} M_s\} \tag{12}$$

where $\{m_1, \ldots, m_n\} \in \{1, \ldots, M_1\} \times \cdots \times \{1, \ldots, M_n\}$, $\mathcal{I}_{i,1} = [\eta_{i,0}, \eta_{i,1}]$, $\mathcal{I}_{i,2} = [\eta_{i,1}, \eta_{i,2}]$, …, $\mathcal{I}_{i,M_i} = [\eta_{i,M_i-1}, \eta_{i,M_i}]$, $\eta_{i,m}$, $m \in \{0, 1, \ldots, M_i\}$ are defined $\eta_{i,n} = \eta_{i,0} + \frac{m(\eta_{i,M_i} - \eta_{i,0})}{M_i}$, $m \in \{0, 1, \ldots, M_i\}$.

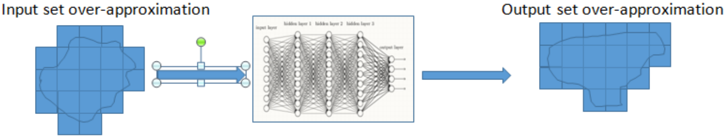
Hyper-Rectangular Sets

$$\mathcal{P}_i = \mathcal{I}_{1,m_1} \times \cdots \times \mathcal{I}_{n,m_n},\ i \in \{1, 2, \ldots, \prod\nolimits_{s=1}^{n} M_s\} \tag{12}$$

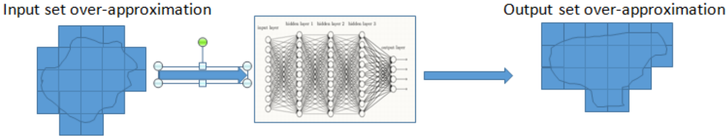
where $\{m_1, \ldots, m_n\} \in \{1, \ldots, M_1\} \times \cdots \times \{1, \ldots, M_n\}$, $\mathcal{I}_{i,1} = [\eta_{i,0}, \eta_{i,1}]$, $\mathcal{I}_{i,2} = [\eta_{i,1}, \eta_{i,2}]$, ..., $\mathcal{I}_{i,M_i} = [\eta_{i,M_i-1}, \eta_{i,M_i}]$, $\eta_{i,m}$, $m \in \{0, 1, \ldots, M_i\}$ are defined $\eta_{i,n} = \eta_{i,0} + \frac{m(\eta_{i,M_i} - \eta_{i,0})}{M_i}$, $m \in \{0, 1, \ldots, M_i\}$.



Reachability of Neural Networks



Reachability of Neural Networks



Reachable Set Estimation (General)

Consider a single layer $\mathbf{u} = \phi(\mathbf{W}\boldsymbol{\eta} + \boldsymbol{\theta})$, if the input set is a hyper-rectangle described by $\mathcal{I}_1 \times \cdots \times \mathcal{I}_{n_\eta}$ where $\mathcal{I}_i = [\underline{\eta}_i, \overline{\eta}_i]$, $i \in \{1, \ldots, n_\eta\}$, the output set can be over-approximated by a hyper-rectangle in the expression of intervals $\mathcal{I}_1 \times \cdots \times \mathcal{I}_{n_u}$, where $\mathcal{I}_i$, $i \in \{1, \ldots, n_u\}$, can be computed by $\mathcal{I}_i = [\underline{u}_i, \overline{u}_i]$, where $\underline{u}_i$ and $\overline{u}_i$ are computed by

$$\underline{u}_i = \min_{\boldsymbol{\eta} \in \mathcal{H}} \phi\left(\sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i\right), \tag{13}$$

$$\overline{u}_i = \max_{\boldsymbol{\eta} \in \mathcal{H}} \phi\left(\sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i\right). \tag{14}$$

Monotonic Assumption

For any two scalars $z_1 \leq z_2$, the activation function satisfies $\phi(z_1) \leq \phi(z_2)$.

Reachability of Neural Networks

Monotonic Assumption

For any two scalars $z_1 \leq z_2$, the activation function satisfies $\phi(z_1) \leq \phi(z_2)$.

LP Formulation

$$
\begin{aligned}
&\min \underline{z}_i \\
\text{s.t.}\quad & \underline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i, \\
& \underline{\eta}_j \leq \eta_j \leq \overline{\eta}_j,\ j \in \{1, \ldots, n_\eta\}, \\
&\max \overline{z}_i \\
\text{s.t.}\quad & \overline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i \\
& \underline{\eta}_j \leq \eta_j \leq \overline{\eta}_j,\ j \in \{1, \ldots, n_\eta\}.
\end{aligned}
$$

Reachability of Neural Networks

Monotonic Assumption

For any two scalars $z_1 \leq z_2$, the activation function satisfies $\phi(z_1) \leq \phi(z_2)$.

LP Formulation

$$\begin{aligned} & \min \underline{z}_i \\ \text{s.t.} \quad & \underline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i, \\ & \underline{\eta}_j \leq \eta_j \leq \overline{\eta}_j,\ j \in \{1, \ldots, n_\eta\}, \\ & \max \overline{z}_i \\ \text{s.t.} \quad & \overline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \omega_{ij}\eta_j + \theta_i \\ & \underline{\eta}_j \leq \eta_j \leq \overline{\eta}_j,\ j \in \{1, \ldots, n_\eta\}. \end{aligned}$$

Solution

$$\underline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \underline{g}_{ij} + \theta_i$$

$$\overline{z}_i = \sum\nolimits_{j=1}^{n_\eta} \overline{g}_{ij} + \theta_i$$

with $\underline{g}_{ij}$ and $\overline{g}_{ij}$ defined by

$$\underline{g}_{ij} = \begin{cases} \omega_{ij}\underline{\eta}_j, & \omega_{ij} \geq 0 \\ \omega_{ij}\overline{\eta}_j, & \omega_{ij} < 0 \end{cases}$$

$$\overline{g}_{ij} = \begin{cases} \omega_{ij}\overline{\eta}_j, & \omega_{ij} \geq 0 \\ \omega_{ij}\underline{\eta}_j, & \omega_{ij} < 0 \end{cases}.$$

For simplicity, we consider a random neural network with two inputs, two outputs and one hidden layer consisting of 7 neurons. Input Set:
$\mathcal{H} = \{\boldsymbol{\eta} \in \mathbb{R}^2 \mid \|\boldsymbol{\eta}\|_\infty \leq 1\}$.

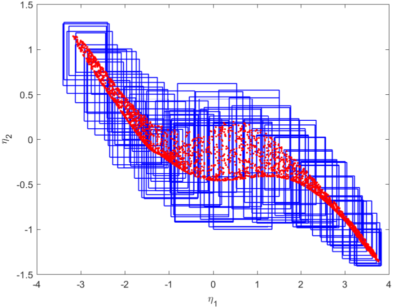


Figure: Output Set Estimation

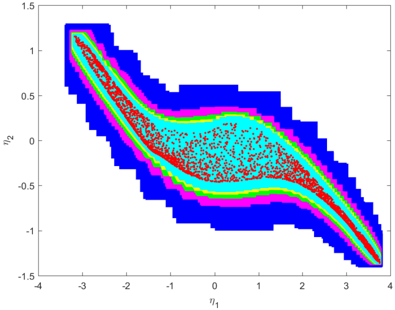


Figure: Output Set Estimations with Different Partition Number

Example

Table: Computation time and number of reachtubes with different partition numbers

Partition Number	Reachtube Number	Computation Time
$M_1 = M_2 = 10$	100	0.062304 seconds
$M_1 = M_2 = 20$	400	0.074726 seconds
$M_1 = M_2 = 30$	900	0.142574 seconds
$M_1 = M_2 = 40$	1600	0.251087 seconds
$M_1 = M_2 = 50$	2500	0.382729 seconds

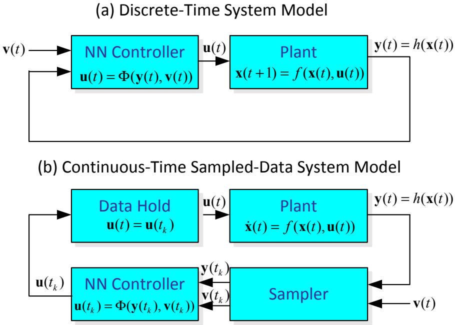


Figure: Illustration for closed-loop systems with neural network controllers

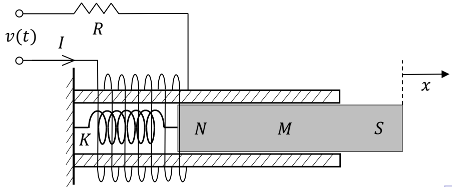
Reachable Set Estimation

- **Plant:**
  Use existing tools for hybrid systems SpaceEX, Flow*, CORA, ...
- **Neural Network Controller:**
  Reachability results of neural networks

State Space Model

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K/M & -W/M \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha/RM \end{bmatrix} v(t) \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \end{cases} \tag{15}$$

where $x_1(t)$ and $x_2(t)$ respectively denote the position and velocity of the magnet bar.



Train neural network controller

- Sampling time 0.1 s
- Model reference control
- Neural network: 1 hidden layer, 7 neurons

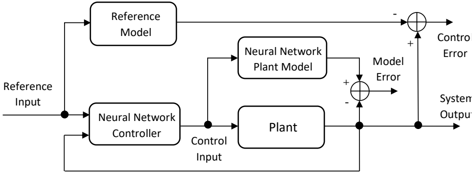


Figure: Model reference control architecture

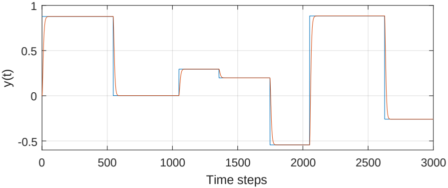


Figure: Performance Test

Initial state set: $\mathcal{X}_0 = \{\|\mathbf{x} - \mathbf{x}_{c,1}\|_\infty \leq 0.1\}$ where $x_{c,1} = [0.9 \ \ -0.9]^\top$.
Reference input set: $\mathbf{r} = \{r \in \mathbb{R} \mid 99 \text{ cm} \leq r \leq 100 \text{ cm}\}$.
Unsafe set: $\neg\mathcal{S} = \{x_1 \in \mathbb{R} \mid x_1 \leq -1.5 \text{ or } x_1 \geq 1.2\}$.

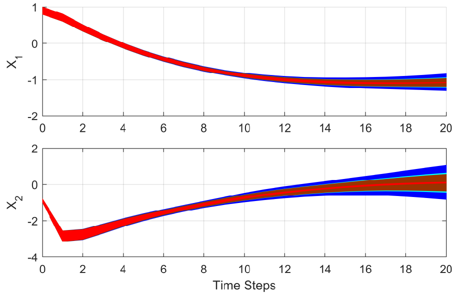


Figure: Flow pipes

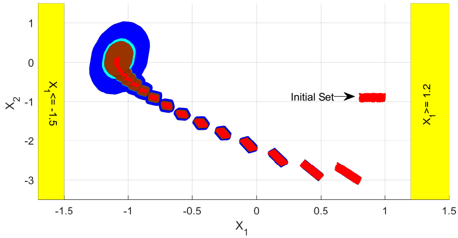


Figure: Reachable sets, system is safe.

Conclusion

- Reachable set estimation for feedforward neural networks by partitioning input space into hyper-rectangles.
- Tighter estimation by finer partition.
- LP formulation and interval solution.
- Reachable set estimation for closed-loop systems.
  - Plant: Existing tools for hybrid systems
  - Neural network controller: Reachability results of neural networks.
- Future work: enhance scalability such as specification-guided methods with an adaptive partition for input space.

Thank you!