Adaptive Neural Replication and Resilient Control Despite Malicious Attacks

Salvatore Giorgi, Firdous Saleheen, Frank Ferrese, Chang-Hee Won

CSNAP Laboratory, Temple University

ISRCS 2012, Salt Lake City, UT

This material is partly supported by the National Science Foundation
AIS Grant, ECCS-0969430

Motivation

- Need for robust, reliable, and resilient controller
  - Resilient controller defined as one which maintains state awareness as well as operational normalcy in response to malicious attacks
  - Robust in terms of compatibility with a wide variety of plant models and able maintain control despite noise and model uncertainty
- We define a malicious attack as one which does one of the following:
  - False Data Injection
    - The attacker modifies the input data to the plant or injects false data
  - Sensor Data Alteration
    - The attacker modifies the output data from the plant
  - Plant Parameter Changes
    - Plant parameters are modified or the entire model of the plant is changed

Outline

- Motivation
- Objective
- Adaptive Neural Control Architecture
- Simulations
- Conclusions

Objective

- To develop a Resilient Model Reference Adaptive Neural Controller
- Model Reference Control Architecture controls the closed-loop output of the plant to match the output of an ideal reference system
  - Here the ideal reference system is independent of the plant
- Neural networks are used to identify the unknown plant and thus little modeling is needed

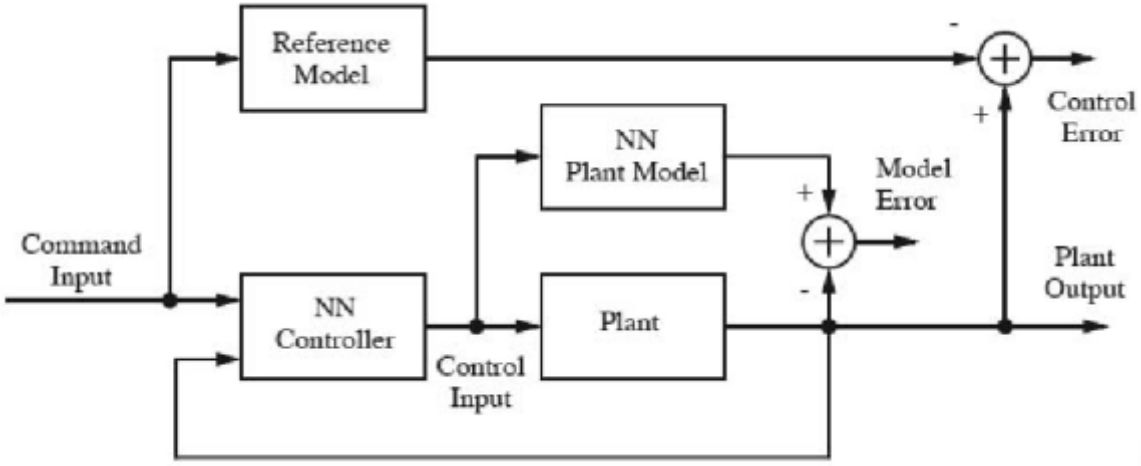
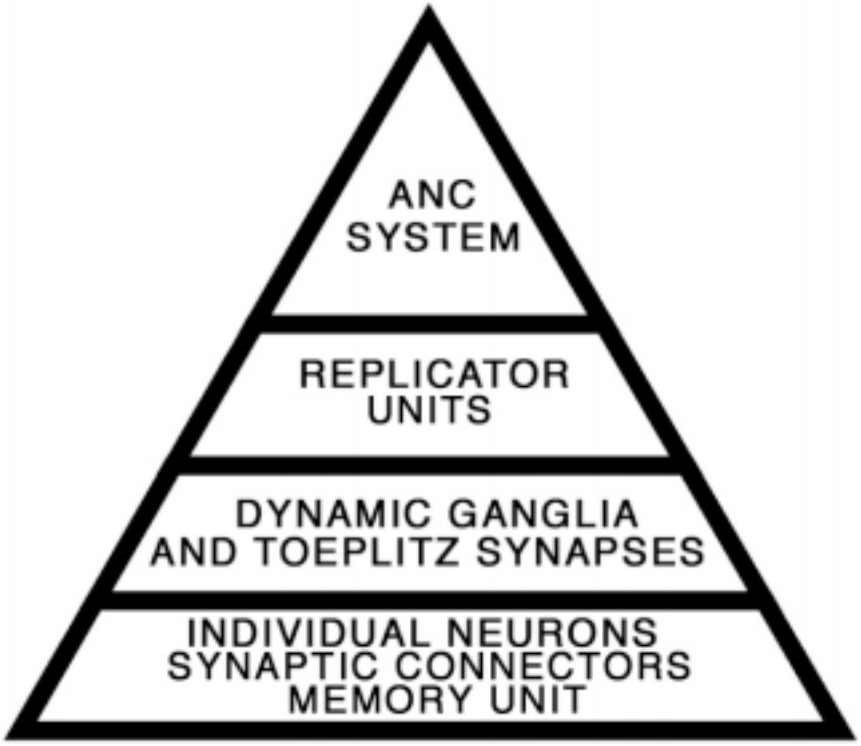


Figure: Model Reference Neural Control Architecture

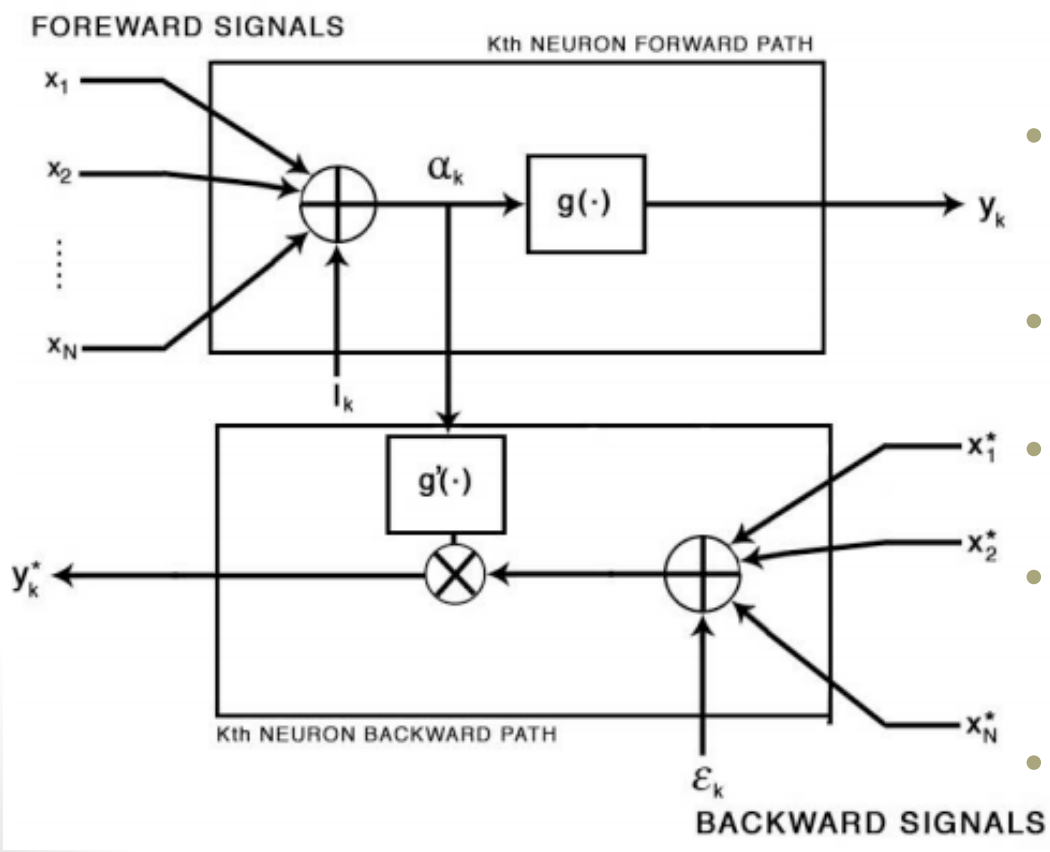
Adaptive Neural Control System

- Hierarchical and modular design gives the system a high level of fault tolerance
- Two separate neural networks are used
  - One replicates the unknown plant
  - The other controls the plant to behave as the ideal reference system
- Two defining characteristics of this neural architecture
  - Time-varying adaptive speed rate
  - Constrained interconnections between neurons impart a temporal ordering on neural network

Hierarchy of ANC System

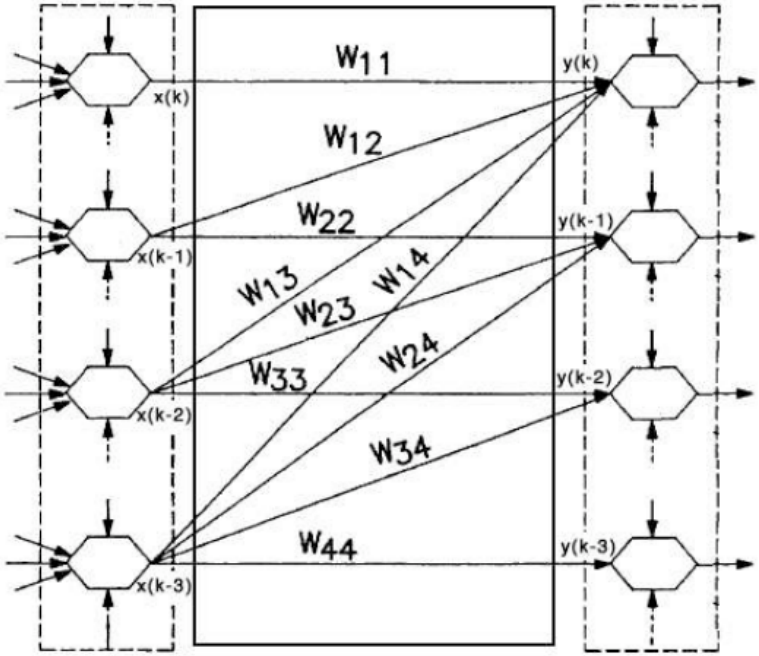


Hierarchy: Neurons and Synaptic Connectors



- Neurons are bi-directional devices with a forward path and a backward path
- Each neuron contains a neural function, which is either a linear or a sigmoid function
- Neurons connected by synaptic connectors
- Synaptic connectors are also bi-directional devices
- Weight associated with synaptic connector and not with neuron
- This allows the neurons to remain static while only the connections adapt

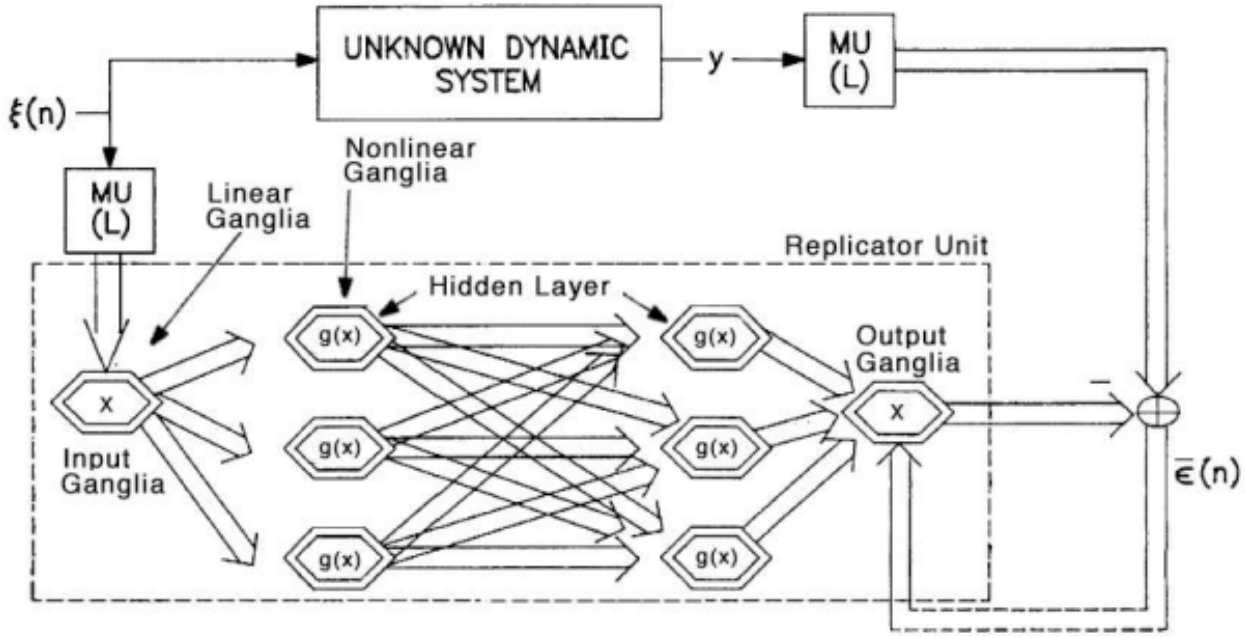
Hierarchy: Ganglia and Toeplitz Synapses



$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ 0 & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{NN} \end{pmatrix}$$

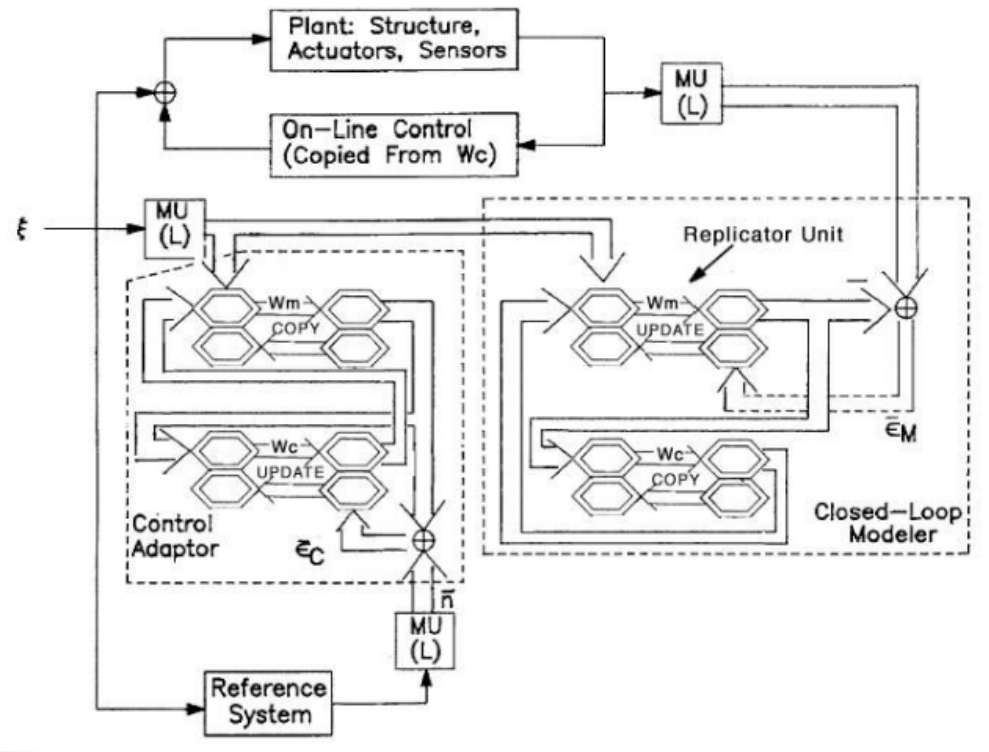
- Groups of neurons are defined as Ganglia
- The position of the neurons determine the age of the data
  - Top level neurons represent current data
  - Lower level neurons represent past data
- Since top level neurons do not feed signals into lower level neurons, past data points do not depend on future inputs
- Groups of synaptic connectors constrained as above are called Toeplitz Synapses
- These weights can be represented by an upper right diagonal weight matrix

Hierarchy: Replicator Unit



- To replicate a system we inject a training signal into the unknown plant and into the neural network
- The error between the plant's output and the neural network's output is then injected into the backward path of the neural network, driving the weight update laws

Hierarchy: ANC System



- The ANC system uses four replicator units
- Two units in the Closed-Loop Modeler
- Two units in the Control Adaptor
- The Closed-Loop Modeler replicated the unknown plant inside the closed-loop
- The Control Adaptor drives the output from the plant to match that of an ideal reference system

ANC Adaptive Update Law

$$W_k(n+1) = W_k(n) + \mu_k(n) U_0 * (\bar{x}_k^*(n) \bar{x}_k^T(n))$$

$$\mu_k(n) = \beta_k F(n)$$

$$U_0 = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

$$F(n) = \frac{P(n)}{A(n)}$$

$$P(n) = L(\frac{1}{2}\|\bar{\varepsilon}(n)\|^2 - J)$$

$$L(\sigma) = \begin{cases} \sigma : \sigma > 0 \\ 0 : \sigma \leq 0 \end{cases}$$

$$A(n) = \sum_{\omega} \|\bar{x}_k^*(n)\|^2 \|\bar{x}_k(n)\|^2$$

- Weight update law contains a time-varying update speed μ
- This update speed depends on the global errors as well as the local forward and backward signals
- β is a constant built into the system and depends on the location of the synapse (linear / nonlinear neuron and control adaptor / closed-loop modeler)

Simulations

- We simulated three types of attacks: False Data Injection, Sensor Data Alteration and Plant Parameter Changes
- Because of the nature of the neural network we assume our plant model to be unknown
- We assume the attack occurs after our plant has been running for sometime and therefore the plant's output already matches that of the ideal reference system
- For simulation purposes we use the follow nonlinear model for our plant:

$$\dot{x}(t) = -f[x(t)] + u(t)$$

$$f[x(t)] = 2x(t) + 0.8x^3(t)$$

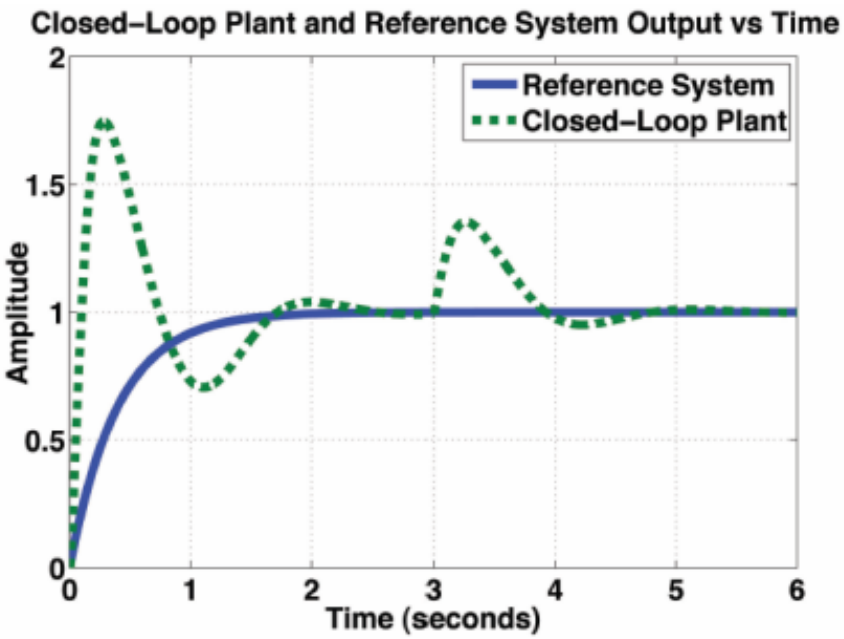
$$y(t) = x(t)$$

- We use the following model for our ideal reference system

$$\dot{x}(t) = -2.5x(t) + 2.5u(t)$$

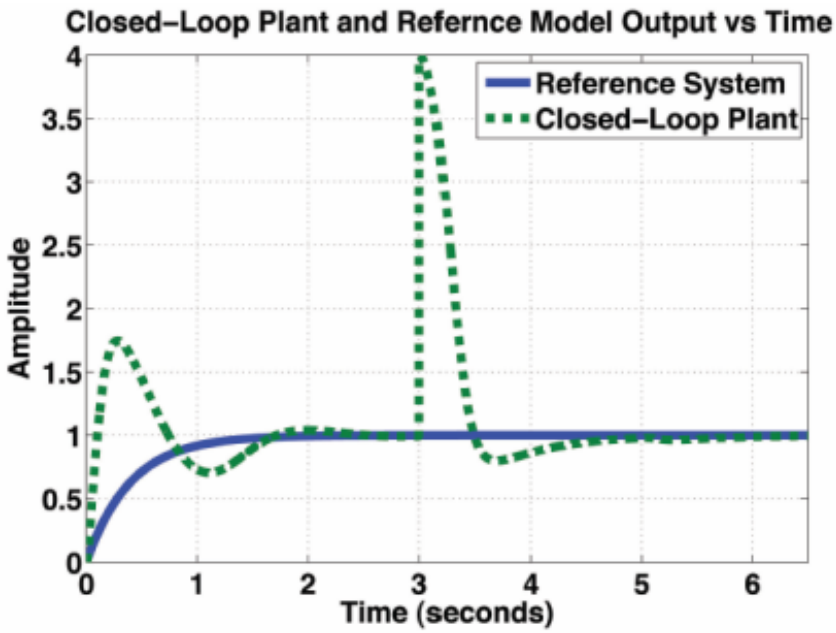
$$y(t) = x(t).$$

Simulations: False Data Injection



- False data is modeled as a step input added to the original plant input signal
- Attack occurs at t = 3 seconds
- Plant's output matches that of the ideal reference system approximately 2 seconds after the attack

Simulations: Sensor Data Alteration



- Sensor Data Alteration is modeled as a step input added to the output of the plant
- Attack occurs at t = 3 seconds
- Plant's output matches that of the ideal reference system approximately 2 seconds after the attack

Simulations: Plant Parameter Changes (Linear Case)

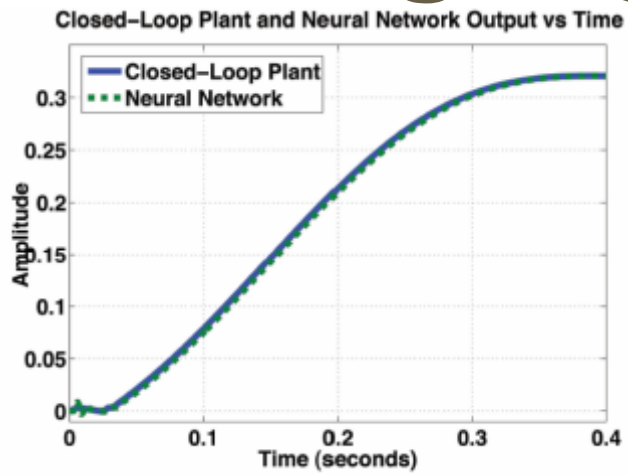


Figure: System Replication

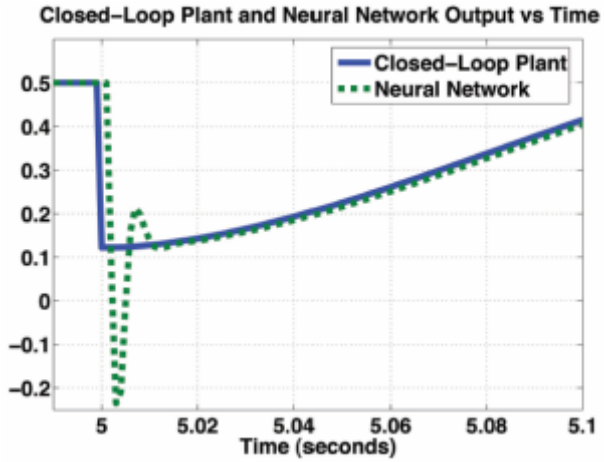


Figure: Attacked Plant Replication

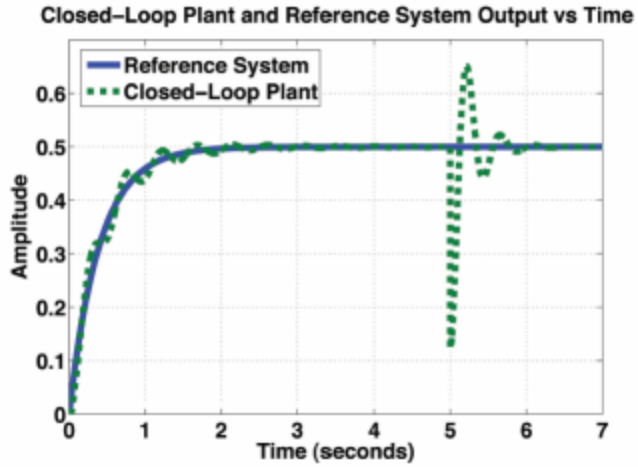
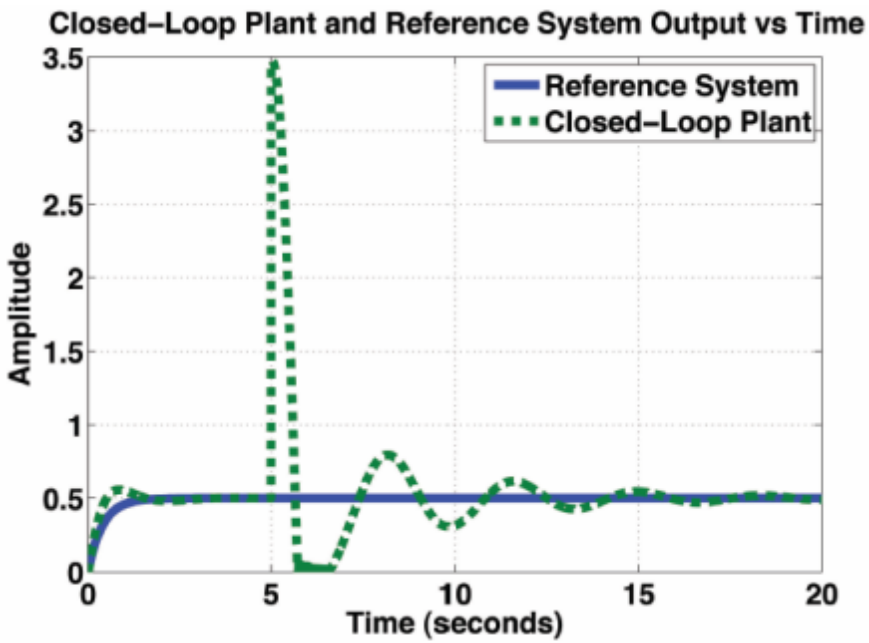


Figure: Control

- Attack plant model: $T(s) = \frac{1}{s+9}$
- Attack occurs at t = 5 seconds
- We see the initial plant replicated in less than 0.1 seconds
- Attacked plant replicated within 0.01 seconds
- Attacked plant controlled within 1 second

Simulations: Plant Parameter Changes (Nonlinear Case)



- Here the attack changes the plant model from a nonlinear model to a completely different nonlinear model

$$\dot{x}(t) = -f[x(t)] + u(t)$$

$$f[x(t), u(t)] = \frac{y(t)}{1 + y^2(t)} + \tanh[u(t)]$$

$$y(t) = x(t)$$

- Attack occurs at t = 5 seconds
- Attacked plant is controlled approximately 12 seconds after attack

Conclusions

- Adaptive Neural Control system is able to identify an unknown nonlinear plant within seconds
- Control of unknown plant usually occurs within 3-5 seconds
- Maliciously attacked plant is identified within seconds
- Maliciously attacked plant output matches that of reference system within 12 seconds for worst case scenario
- Setting learning rate constants too high can result in highly oscillatory behavior
- Setting learning rate constants too low results in longer convergence times
- We have found a broad range of values for the learning rate constants that give good settling times without much oscillations for a wide variety of plant models

Questions?