ADAPTIVE NEURAL CONTROL OF A GIMBALED LASER TARGETING SYSTEM WITH RESILIENT METRICS

M.S. Thesis Proposal

By
Salvatore Giorgi
salvatore.giorgi@temple.edu

Advisory Chair: Dr. Chang Hee-Won
Committee Members: Dr. John Helferty
Dr. Dennis Silage

Motivation

- Need for robust, fault tolerant, and resilient controllers for highly complex, interconnected systems
  - Robust in terms of the operation of a system under a given range of perturbations or disturbances
  - Fault tolerant is the ability to execute specified algorithms correctly regardless of hardware failures, total system flaws, or program fallacies
  - Resilient controller defined as one which maintains state awareness as well as operational normalcy in response to anomalies, unexpected or malicious
- We define a system anomaly as one of the following:
  - Plant Parameter Changes
    - Plant parameters are modified or the entire model of the plant is changed
  - Inter-system Latencies
    - Complex interconnected systems contain multiple, often unknown, latencies
    - Latencies could result from unexpected failures or attacks on the plant
  - 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

Outline

- Motivation
- Objective
- Contributions
- Background Material
  - Neural Networks
  - Model Reference Adaptive Control
  - Resilient Control
  - Laser Targeting System
- Adaptive Neural Control (ANC) System
- Simulations and Hardware Implementation
- Conclusion and Future Work

Objective

- The ANC system is a neural network controller set within a Model Reference Control architecture
- First proposed in the 1990's by D. C. Hyland
- It was first tested in hardware before any analytical results were completed
- Thus, we propose to study resiliency through hardware implementation
- Because of the computational complexity of the ANC system, proper hardware implementation must exploit certain parallelisms within a neural network
- We propose to implement ANC system in hardware with an FPGA

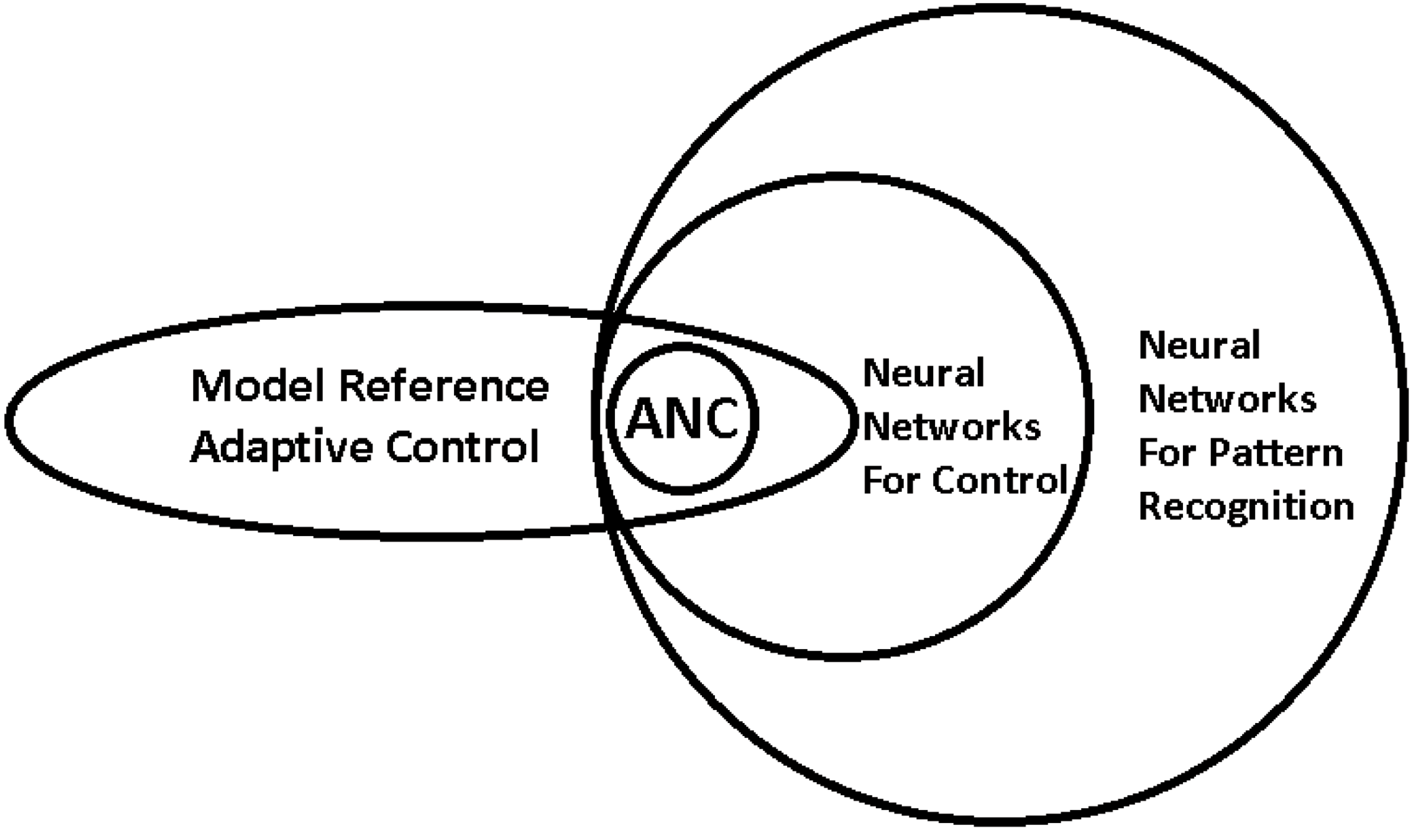
Contributions

- Develop a software implementation of the ANC system in Matlab / Simulink
- Apply controller to laser targeting test bench via sequential processor
- Develop hardware model of ANC system in Xilinx System Generator / Simulink
- Apply controller to test bench via FPGA
- Examine resiliency to system anomalies: plant parameter changes, inter-system latencies, sensor data alteration, and false data injection
- Resiliency will be determined through multiple resilient metrics

Background

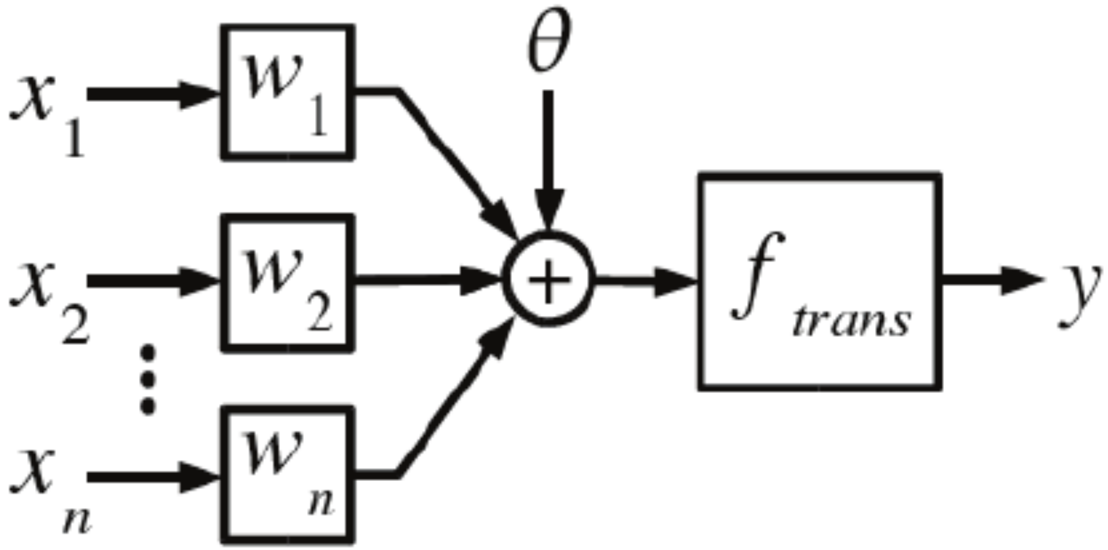
- Neural Networks for System Identification and Control
- Model Reference Adaptive Control
- Resilient Control
  - Definition
  - Resiliency vs Robustness / Adaptiveness / Fault-Tolerance
  - Resiliency Curve
  - Resilient Metrics
- Laser Targeting System
  - Test Bench
  - Linearized Plant Model

Neural Networks



The ANC system sits within the intersection of Neural Network Control and Model Reference Adaptive Control

Neural Networks: Neurons

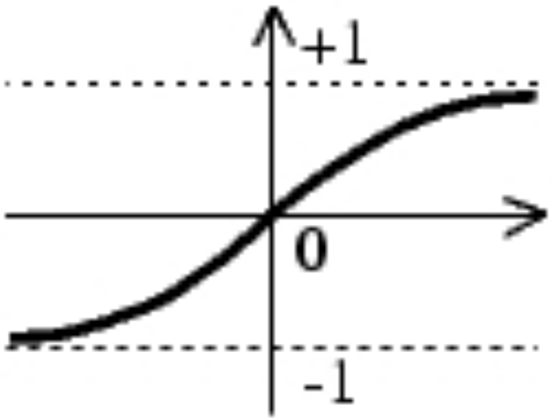


General Neuron

- Arbitrary number of inputs / single output
- Inputs are multiplied by weights and summed with a bias signal
- Sum is propagated to output via neural function

Neural Networks: Neural Functions

$$y = f_{trans}\left(\sum_{i=1}^{n} x_i w_i + \theta\right)$$

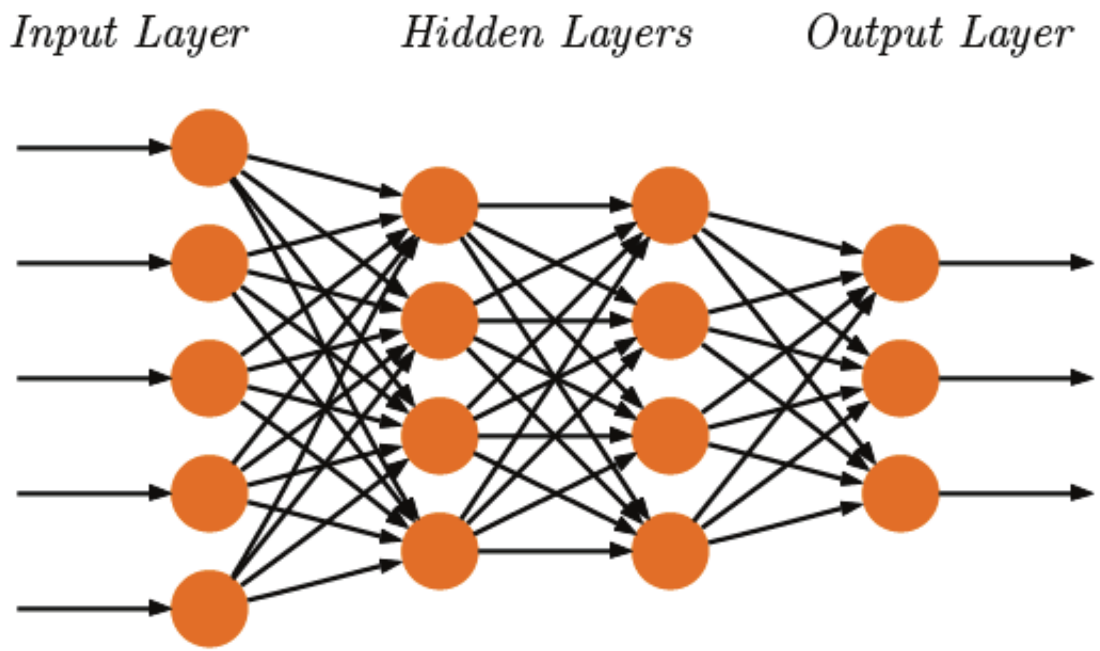


Tansig Function

Neural Function

- **Linear Function**: usually the identity map
- **Threshold or Hard Limit Function**: gives a binary output
- **Sigmoid Function**: bounded, monotone, continuous, and differentiable function

Neural Networks: Layers



Layers

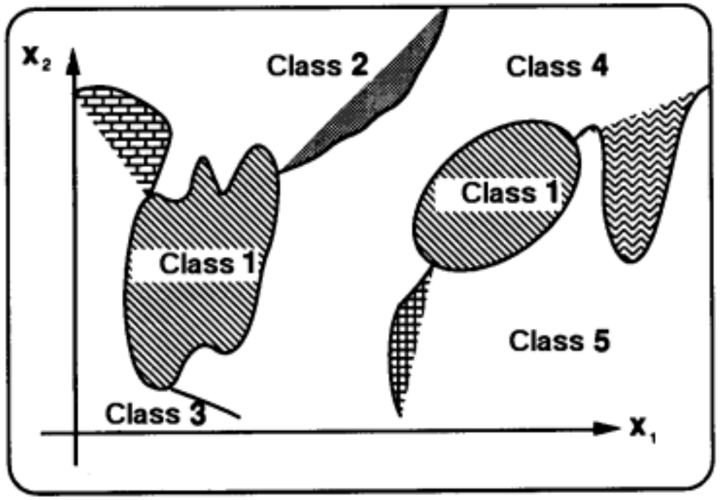
- Simplest neural networks consist of an input and output layer
- Most neural networks contain at least one hidden layer
- The ANC system used consists of a linear input layer, a single nonlinear hidden layer, and a linear output layer

Neural Networks: Learning

Learning

- Two classes of learning: Supervised and Unsupervised
- **Supervised**
  - Trained via input / output pairs
  - Difference between current output and desired output drives the learning process
- **Unsupervised**
  - Trained via input only, since output is not known in advance
  - Neural Network autonomously reconfigures to classify the input

Neural Networks: Back Propagation



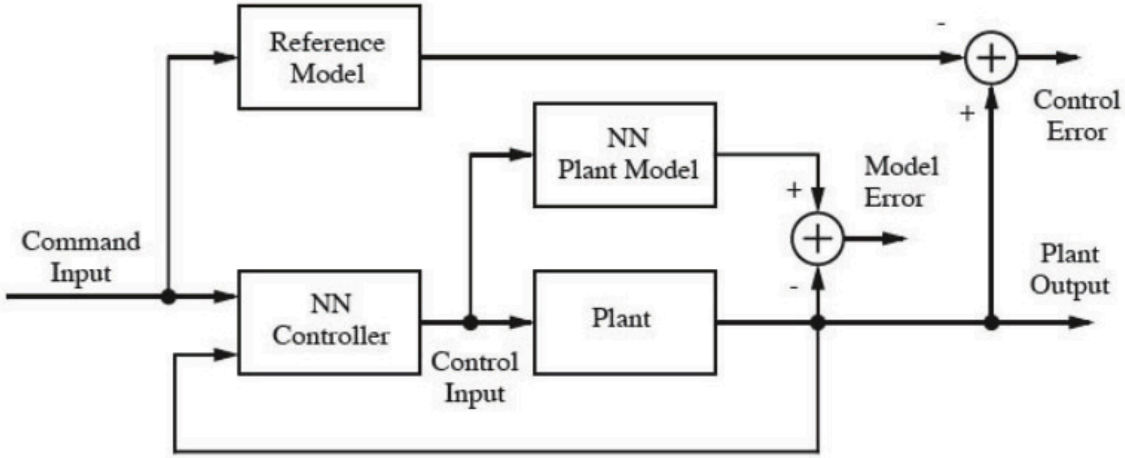
Back Propagation

- To deal with complex, disjointed classification areas, back propagation was introduced by Werbos in 1974
- Hidden layers are needed for this type of classification
- Hidden layers can only be trained through back propagation
- Error is propagated throughout the network, with each neuron receiving an error signal proportional to that neurons contribution to the output

Neural Networks: Parallelisms

- **Layer Parallelism**
  - Different layers can be processed in parallel
  - Less significant than other parallelisms since each layer contains tens of neurons
- **Training Parallelism**
  - Multiple training sessions can be run in parallel
  - Of medium importance since this results in hundreds of neural processes executing simultaneously
- **Node Parallelism**
  - Individual neurons processed in parallel
  - Most important parallelism, as other parallelisms follow
  - Neural networks often consist of thousands to millions of neurons, and, therefore, this is difficult to obtain
- **Weight Parallelism**
  - Weights are updated in parallel

Model Reference Adaptive Control



- Desirable dynamic characteristics of the plant are specified in a reference model
- Input / adaptable plant parameters are changed so that the plant's output matches the reference's output
- Two independent neural networks are used
  - One replicates the plant
  - One controls the plant

Resilient Control

- Definition
- Resiliency vs Robustness / Adaptiveness / Fault-Tolerance
- Resiliency Curve
- Resilient Metrics

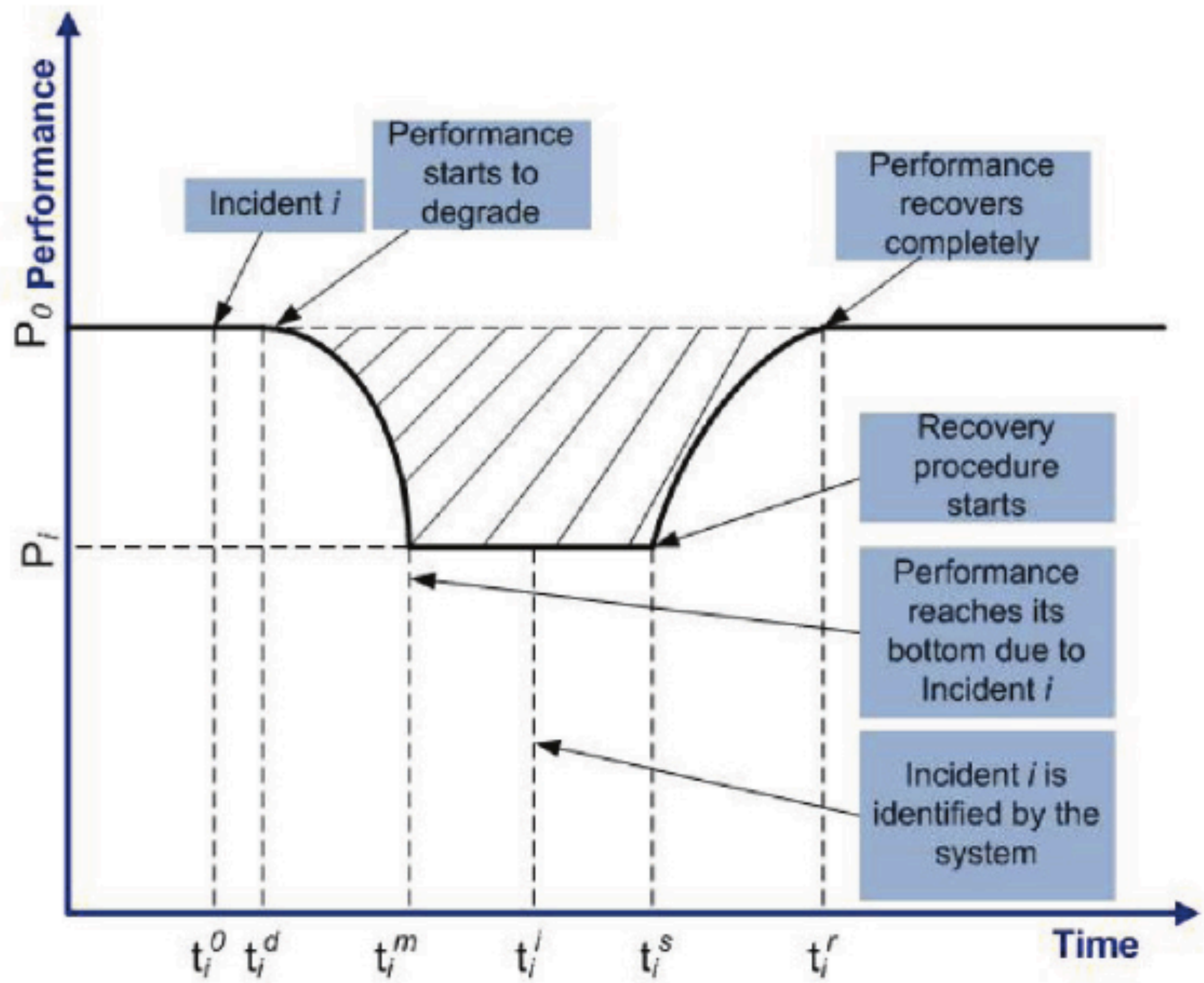
Resilient Control: Definition

- **Resiliency** is defined as the capacity of a control system to maintain state awareness and to proactively maintain a safe level of operational normalcy in response to anomalies
- A **resilient control system** should protect stability, efficiency, and security
- A **resilient control system** is defined as one that is designed to operate in a way that
  - The incidence of undesirable incidents can be minimized
  - Most of the undesirable incidents can be mitigated
  - Adverse impacts of undesirable incidents can be minimized
  - It can recover to normal operation in a short time

Resiliency vs Robustness / Adaptiveness / Fault-Tolerance

- **Robustness**: the ability to maintain satisfactory stability or performance characteristics in the presence of all conceivable system parameters
- **Fault-Tolerance**: the ability of a controlled system to maintain control objectives, despite the occurrence of a fault (defect in sensor, actuator, etc.)
- **Adaptiveness**: ability of the controller to automatically adjust in real time, in order to maintain a desired level of control performance
- None of the above definitions consider how quickly a control system recovers to operational normalcy
- Thus, resiliency is a superset of all of the above properties

Resilient Control: Resiliency Curve



Resilient Control: Metrics

- **Performance Degradation**: maximal performance degradation due to incident *i* (P_0 is the original system performance, P_i is the minimum performance due to the incident)

$$P_i^d = P_0 - P_i$$

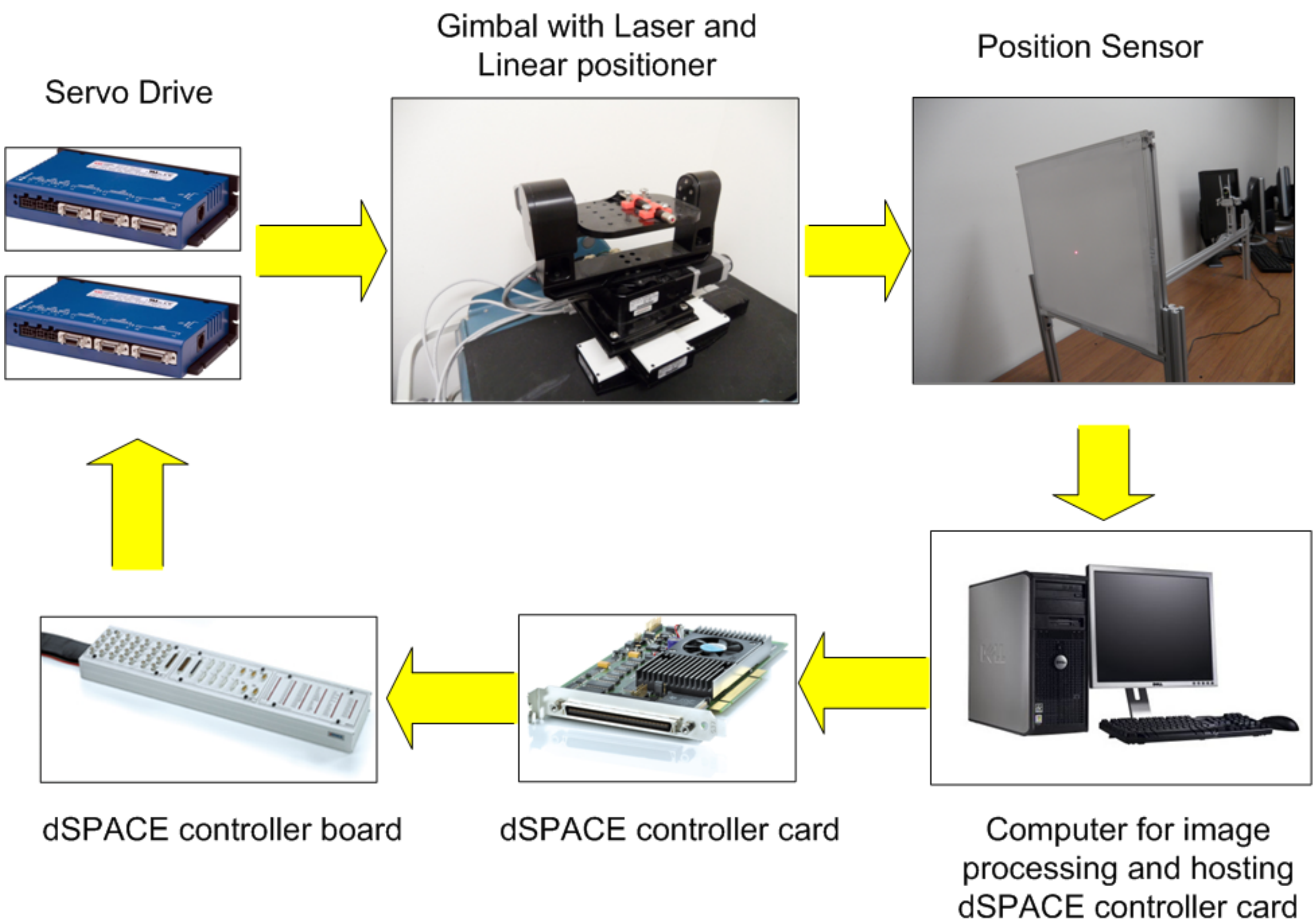
- **Protection Time**: the time that the system can withstand the incident *i* without performance degradation

$$T_i^p = t_i^d - t_i^0$$

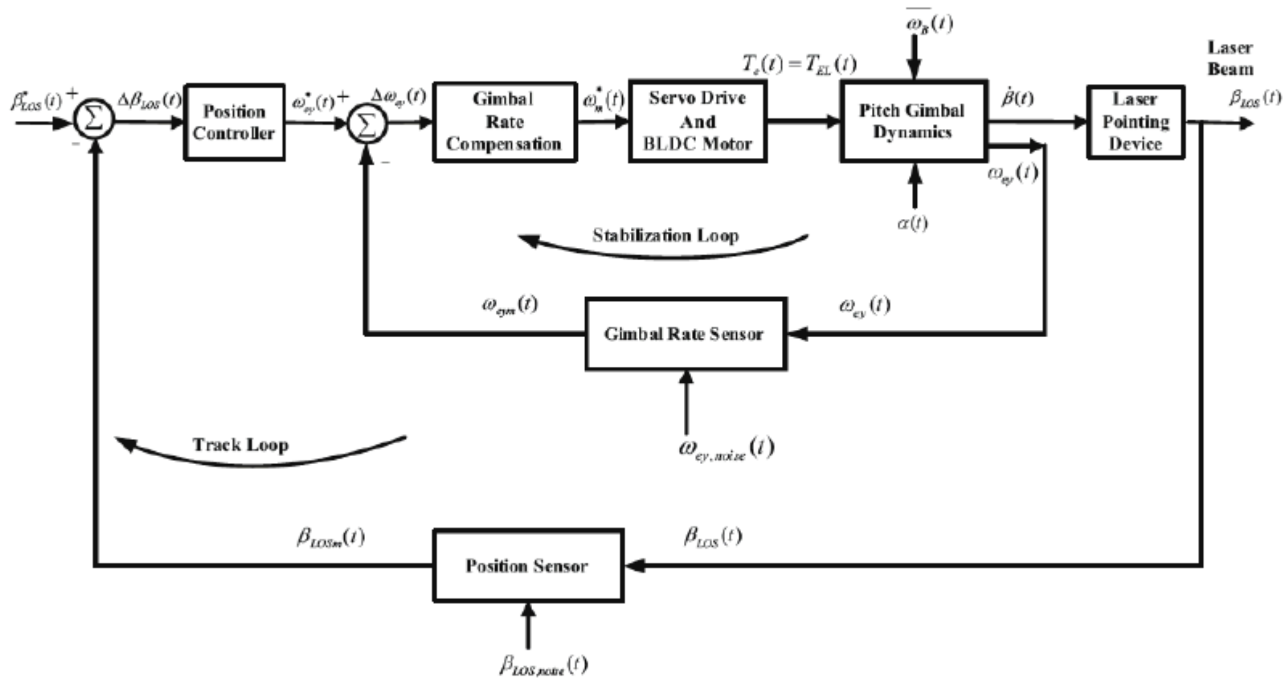
- **Degrading Time**: the time that the system reaches its performance bottom

$$T_i^d = t_i^m - t_i^0$$

Laser Targeting System: Test Bench



Laser Targeting System: Plant Model



- **Track Loop**: maintains laser point at a specified target
- **Stabilization Loop**: maintains the line of sight of laser in a fixed orientation despite disturbances
- **Input**: Pitch line of sight angle command
- **Output**: Pitch line of sight angle

Laser Targeting System: Plant Model

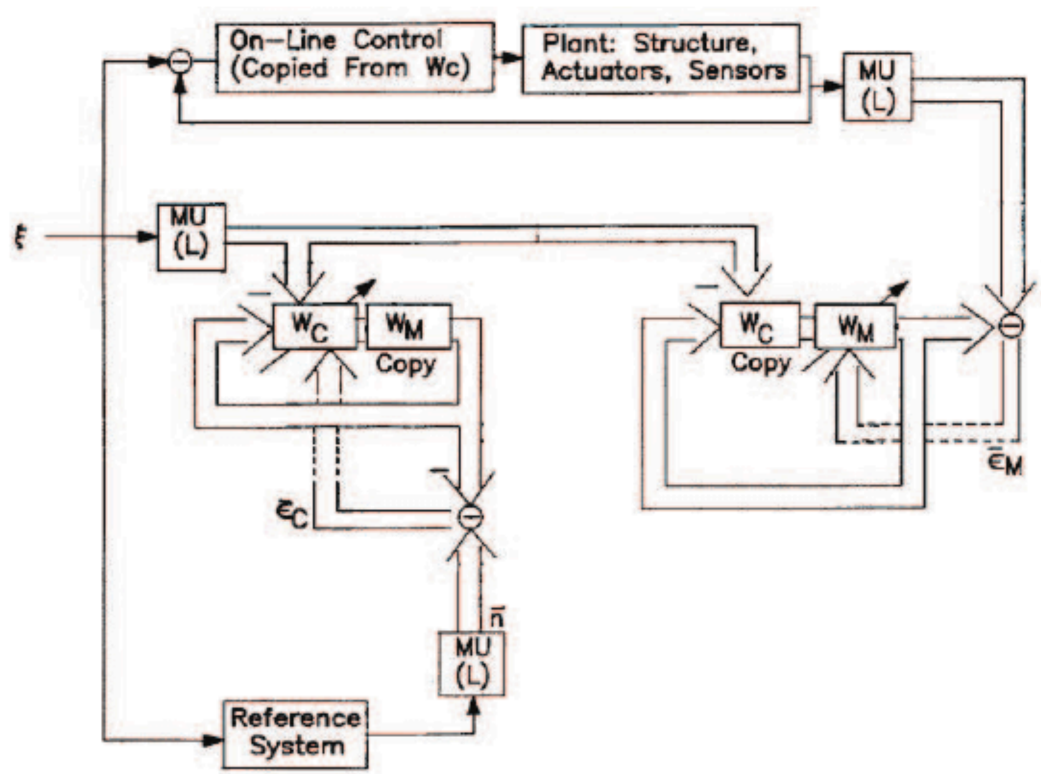
$$T_P(s) = \frac{K_i}{\tau_i s + 1} \times \frac{k_b N s}{J_{ey} s^2 + K_{ef} s + K_{e\omega}}$$

Linearized Pitch Gimbal Model

- J_{ey}: gimbal moment of inertia
- K_{ef}: friction coefficient
- $K_{e\omega}$: cable constraint coefficient
- τ_i: time constant of the reduced current control loop
- N: gear ratio
- k_b: flux linkage
- K_i: gain of the reduced current control loop

ANC System

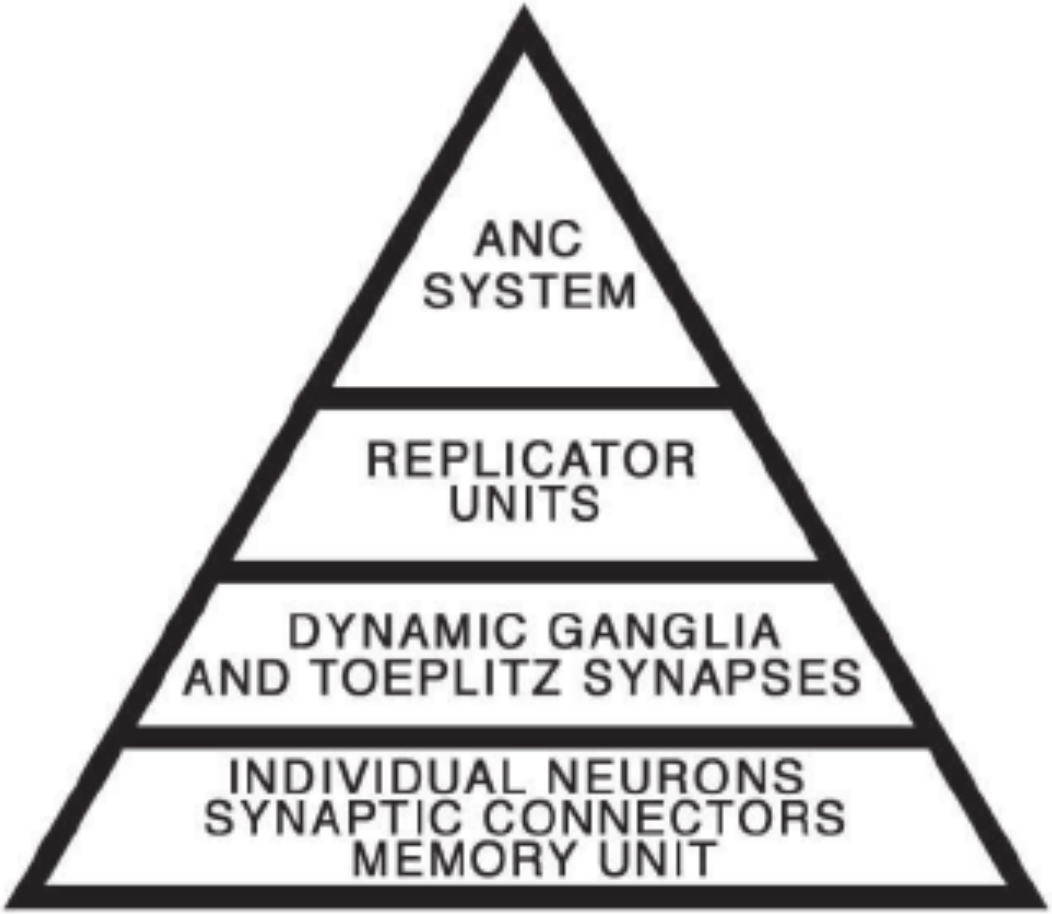
- Hierarchy
- Memory Unit
- Individual Neuron
- Synaptic Connector
- Dynamic Ganglia
- Replicator Unit
- Controller
- Weight Update Law
- Resiliency



ANC 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

ANC System: Hierarchy

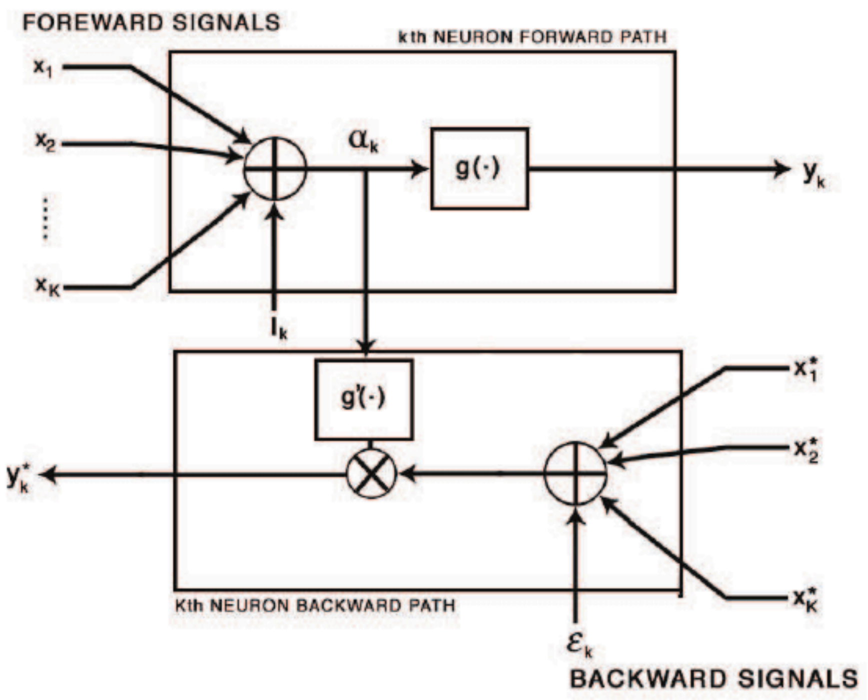


ANC System: Memory Unit

$$\bar{\phi}(n) = \begin{bmatrix} \phi(n) \\ \phi(n-1) \\ \vdots \\ \phi(n-L-1) \end{bmatrix}$$

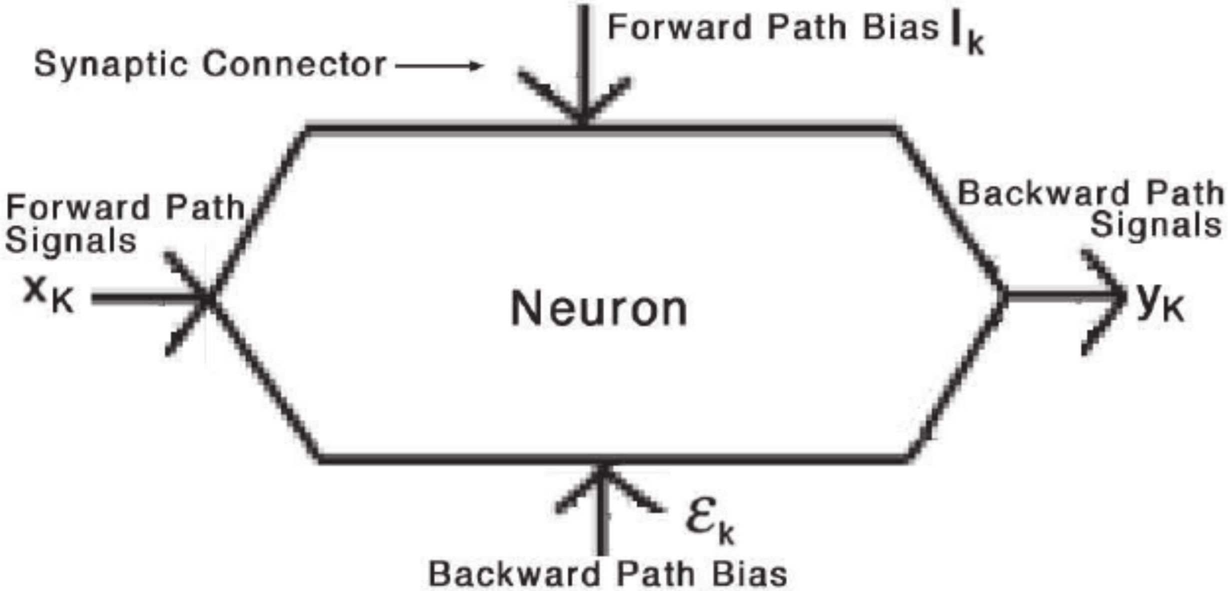
- Takes a scalar time-series input
- Produces an *L*-dimensional vector consisting of
  - Current input
  - *L*-1 delayed values
- Bar notation denotes a column vector with *L* past signal values

ANC System: Individual Neuron



- 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
- Derivative of neural function is multiplied by sum of backward path signals

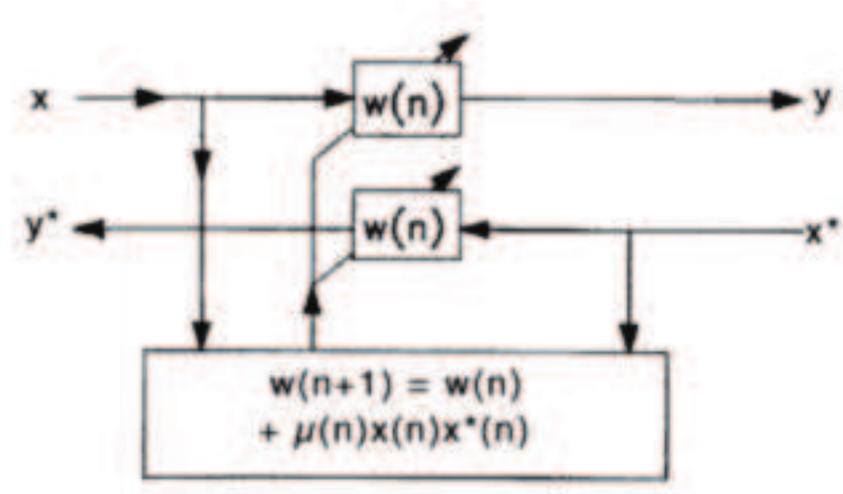
ANC System: Simplified Neuron



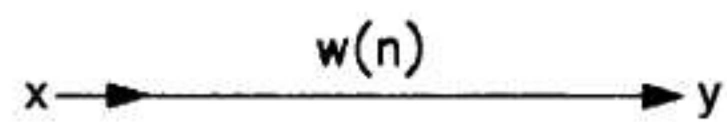
- Hexagon contains everything from previous image
- Only forward paths are shown, backward path signals are implicit
- Location of signal denotes signal type

ANC System: Synaptic Connector

- Connects the output of one neuron to the input of another neuron
- Bi-directional devices
- Weight associated with synaptic connector and not with neuron
- This allows the neurons to remain static while only the connections adapt
- Weight is the same for both forward and backward path signals



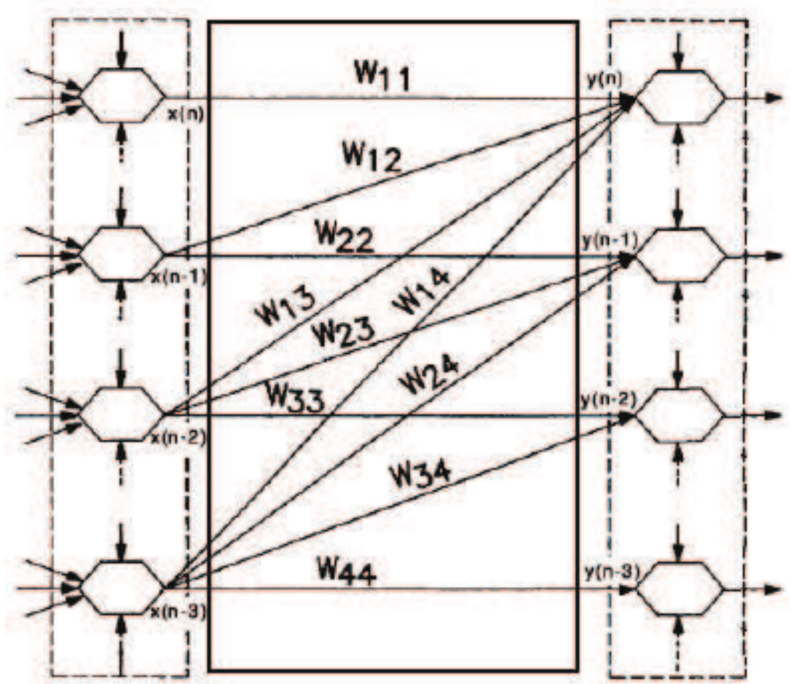
Explicit Synaptic Connector



Simplified Synaptic Connector

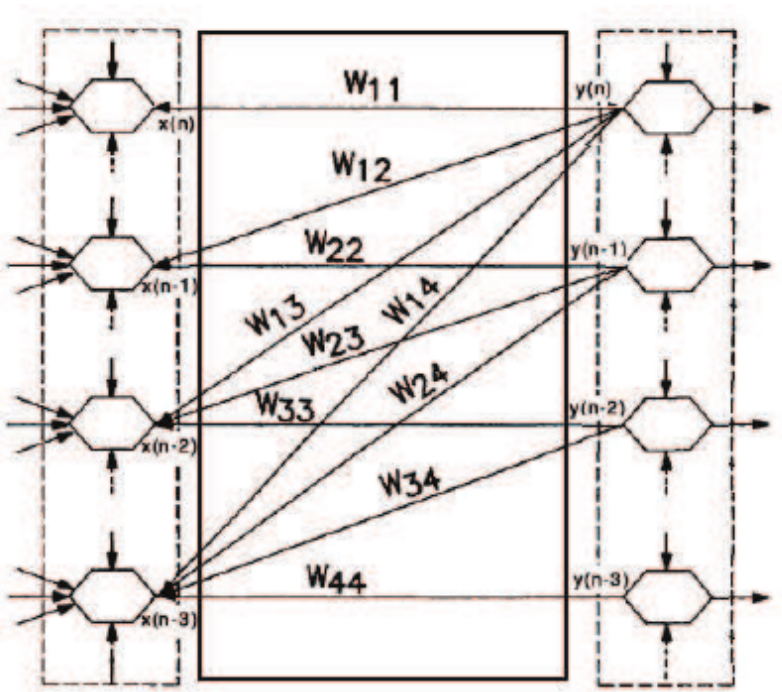
ANC System: Dynamic Ganglia

- 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



$$\begin{bmatrix} y(n) \\ y(n-1) \\ y(n-2) \\ y(n-3) \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ 0 & w_{22} & w_{23} & w_{24} \\ 0 & 0 & w_{33} & w_{34} \\ 0 & 0 & 0 & w_{44} \end{bmatrix} \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ x(n-3) \end{bmatrix}$$

ANC System: Dynamic Ganglia



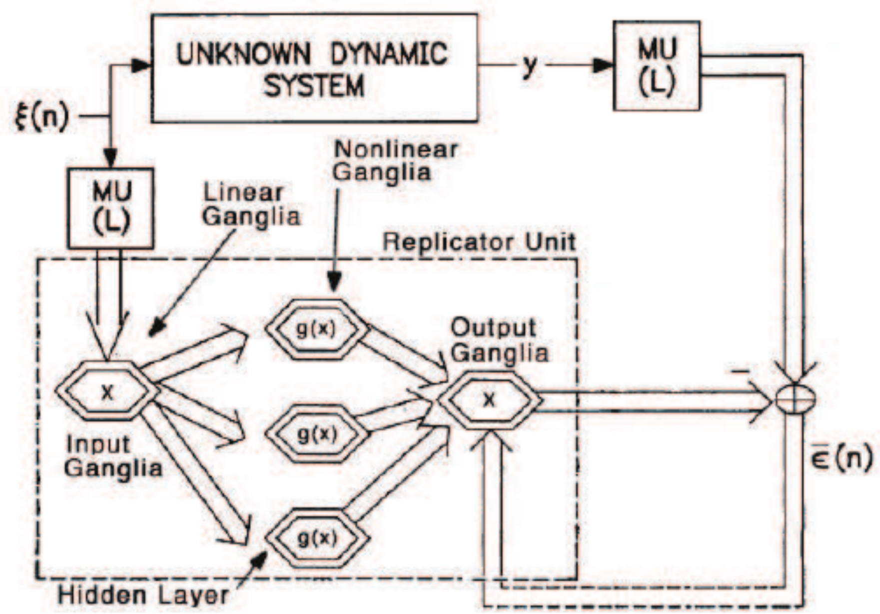
- All synapses are bi-directional
- Backward path of the Toeplitz synapse is constained by the transpose of the forward path weight matrix

$$\begin{bmatrix} y^*(n) \\ y^*(n-1) \\ y^*(n-2) \\ y^*(n-3) \end{bmatrix} = \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ w_{12} & w_{22} & 0 & 0 \\ w_{13} & w_{23} & w_{33} & 0 \\ w_{14} & w_{24} & w_{34} & w_{44} \end{bmatrix} \begin{bmatrix} x^*(n) \\ x^*(n-1) \\ x^*(n-2) \\ x^*(n-3) \end{bmatrix}$$

ANC System: Simplified Ganglia

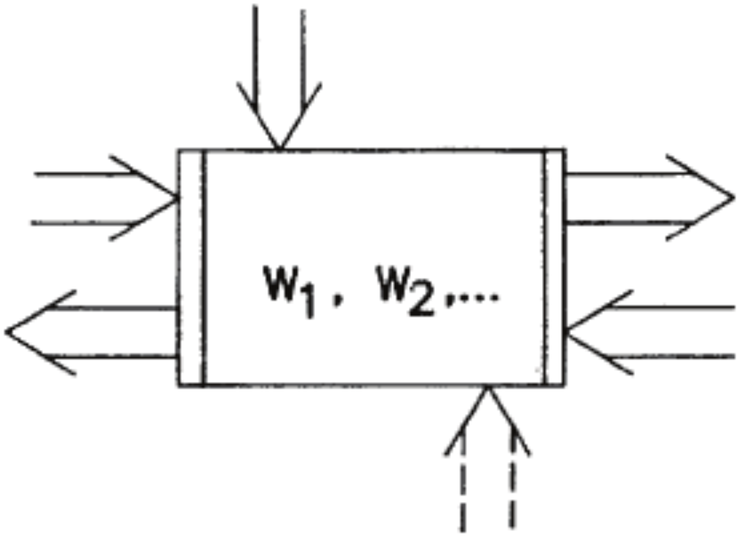
- Double hexagon contains everything from previous image
- Thick arrow denotes Toeplitz synapses
- Only forward paths are shown, backward path signals are implicit
- Location of signal denotes signal type

ANC System: 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

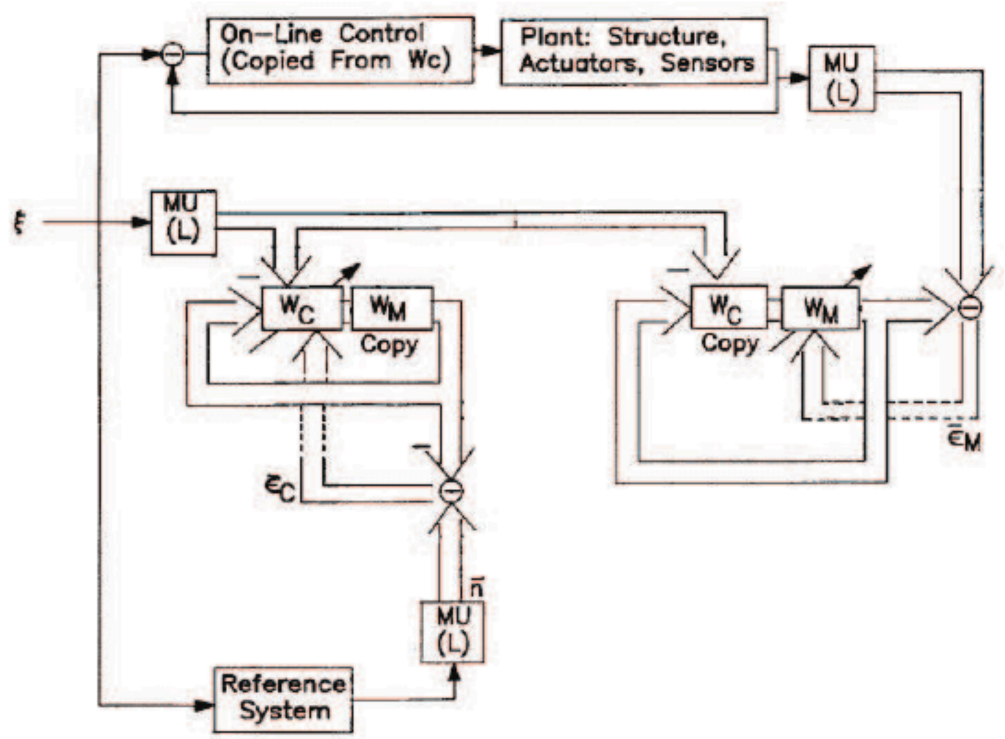
ANC System: Simplified Replicator Unit



- Square contains everything within dotted line in previous image
- Sometimes only forward paths are shown
- Location of signal denotes signal type

ANC System: Controller

- 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 System: Weight Update Law

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

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

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

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

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

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

- 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)

ANC System: Resiliency

- We simulated three types of attacks: Plant Parameter Changes, False Data Injection, and Sensor Data Alteration
- 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

Type of Attack	Recovery Time T_r, (s)
Plant Parameter Change	11
Sensor Data Alteration	3
False Data Injection	2

- We use the following 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

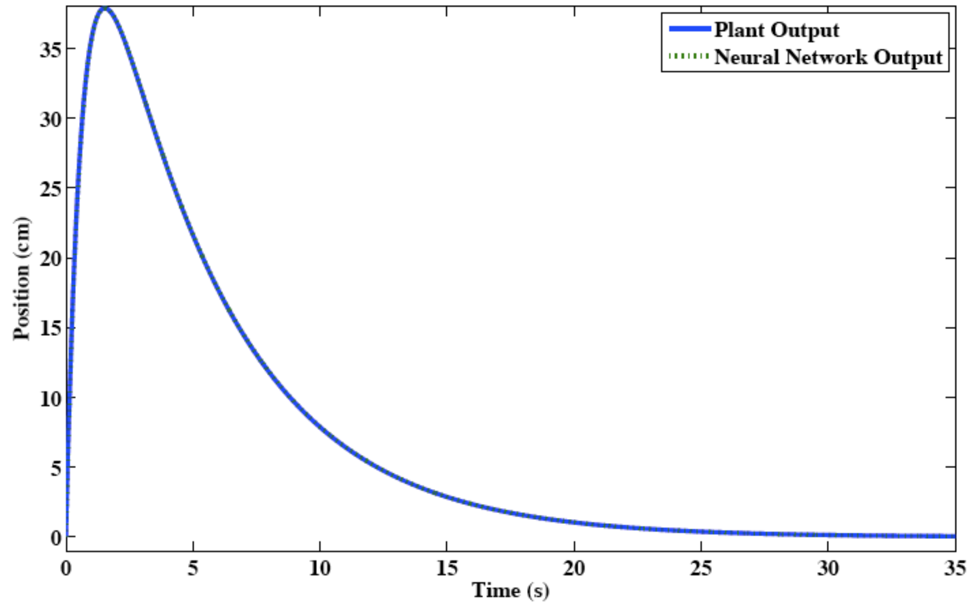
$$\dot{x}(t) = -2.5x(t) + 2.5u(t)$$
$$y(t) = x(t).$$

Simulation and Hardware Implementation

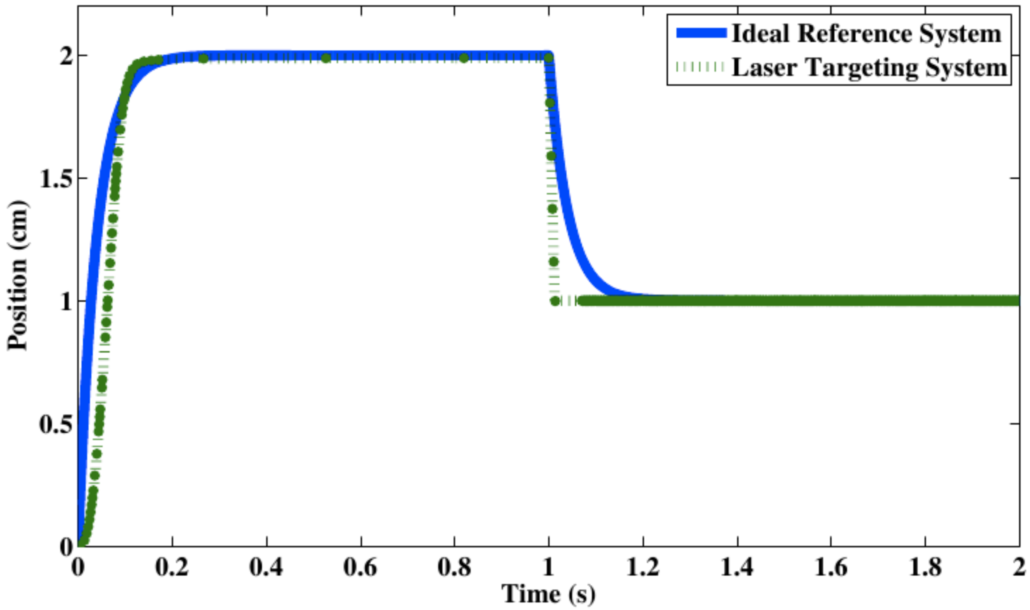
- Simulations
  - System Replication
  - Control via ANC system
- Hardware Implementation
  - Disturbance
  - System Replication
  - Control via ANC system
  - Control via PID controller

Simulations: System Replication

- A unit step input is applied at t = 0
- System is replicated almost instantaneously
- Simulation values
  - 3 neurons per ganglia
  - Initial weight values = 10^{-6}
  - $\alpha = 0.1$
  - $J = 10^{-8}$
  - Sample time = 10^{-6} s



Simulations: Control



- The input is

$$u(t) = \begin{cases} 2, & \text{if } 0 \le t \le 1 \\ 1, & \text{if } t \ge 1 \end{cases}$$

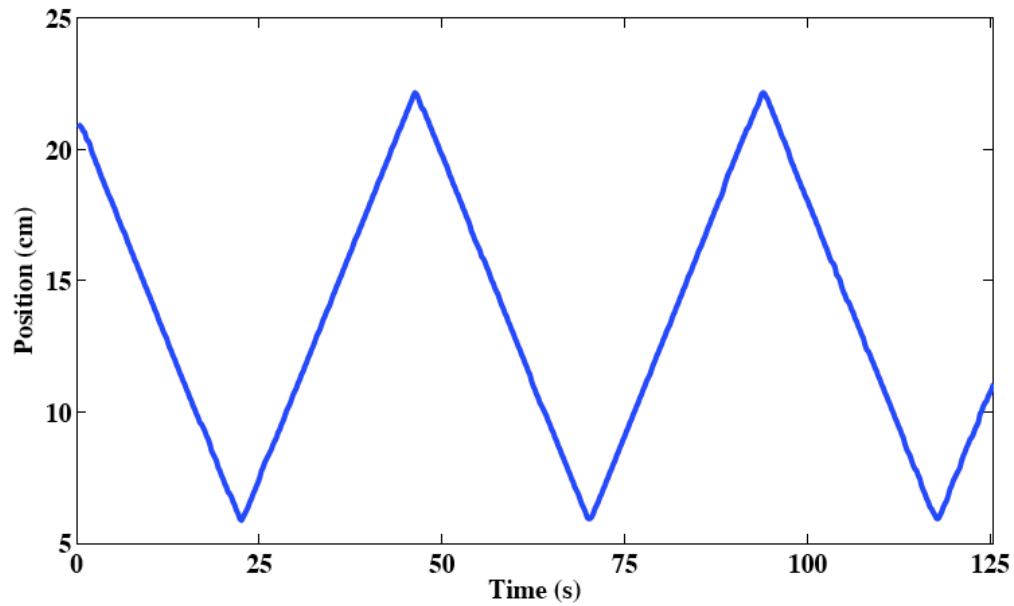
- The ideal reference is

$$T_I(s) = \frac{25}{s+25}.$$

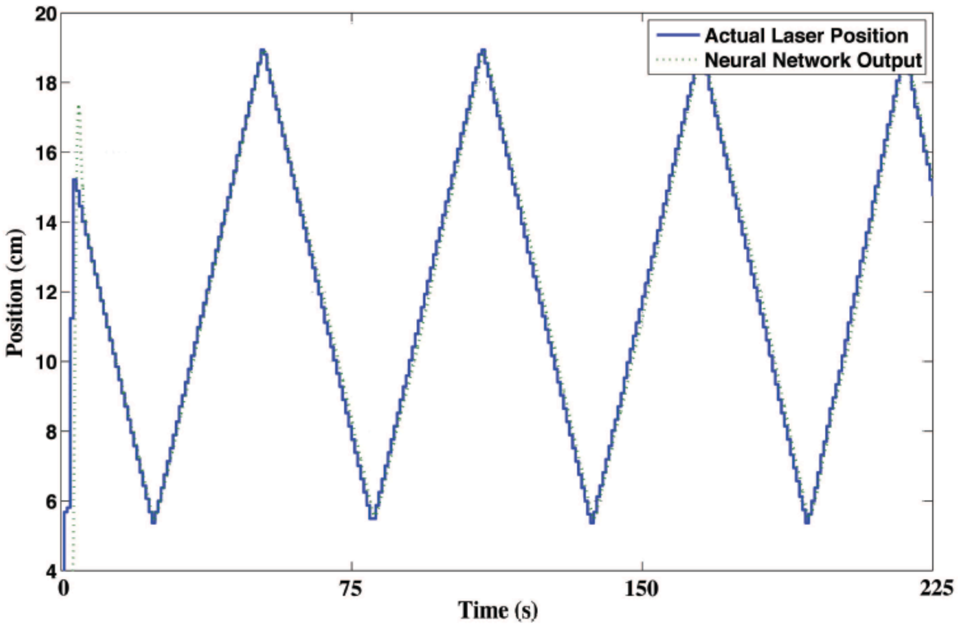
- Simulation values
  - 3 neurons per ganglia
  - Initial weight values = 10^{-6}
  - $\alpha = 0.001$
  - $\beta_C = 0.03$
  - $\beta_M = 0.07$
  - $J = 10^{-8}$
  - Sample time = 10^{-6} s

Hardware: Disturbance

- Disturbance is the same for each experiment
- Gimbal sits on Newmark 5” linear stage mover
- Disturbance speed = 50,000 counts / sec

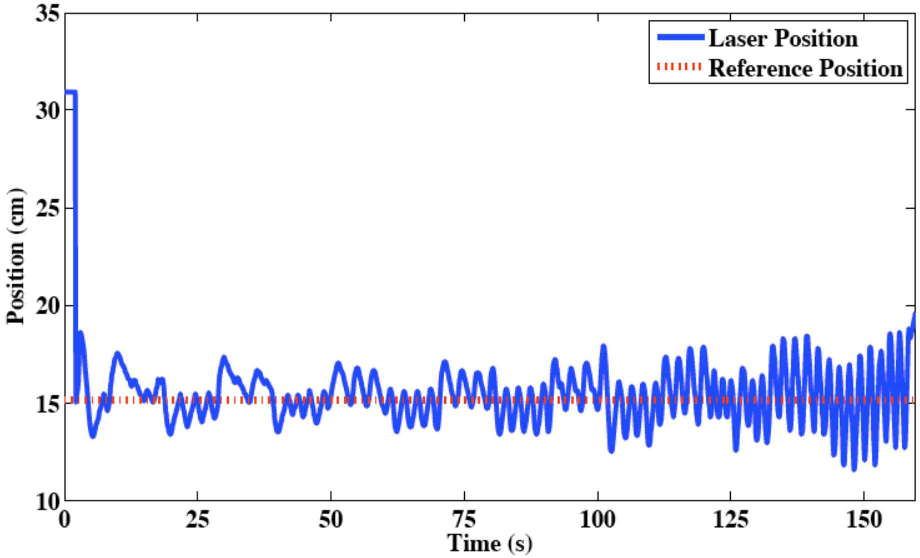


Hardware: System Replication



- System Replication was performed on single gimbal axis
- System is replicated after approximately 10 s
- Experimental values
  - 5 neurons per ganglia
  - Initial weight values = 10^{-6}
  - $\alpha = 0.01$
  - $J = 10^{-8}$
  - Sample time = 0.0001 s

Hardware: ANC Control

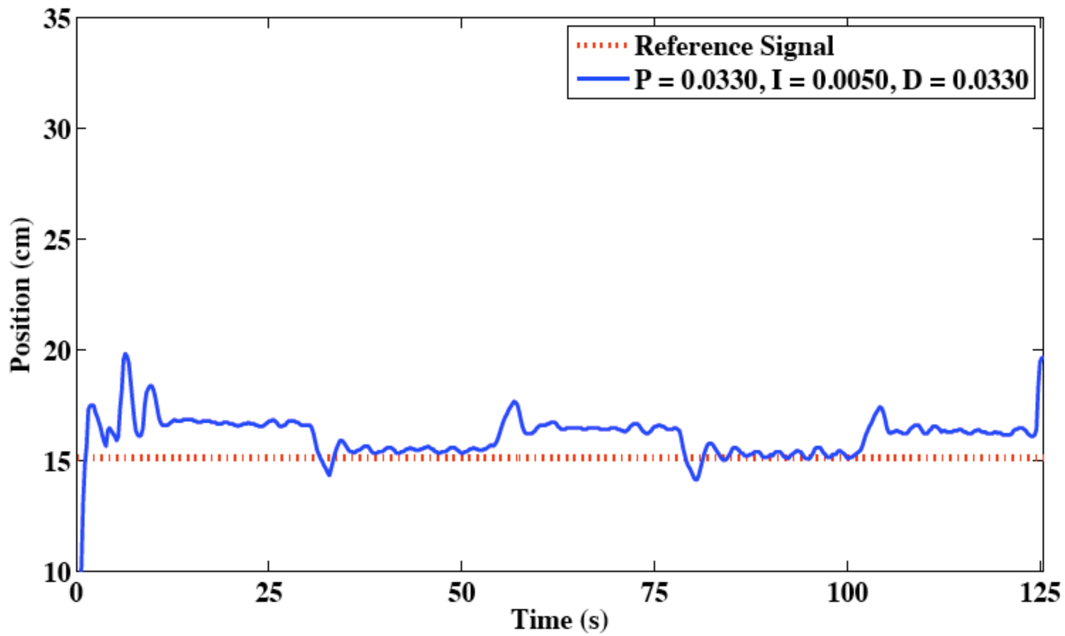


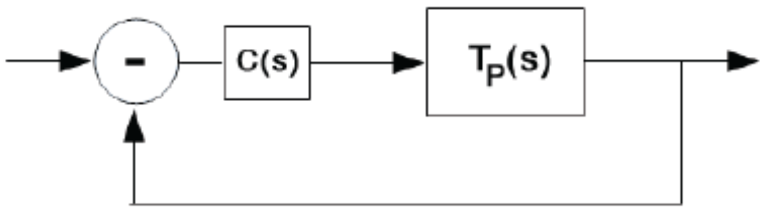
- Control was performed on single gimbal axis
- Laser begins to oscillate significantly after 100 s
- Experimental values
  - 5 neurons per ganglia
  - Initial weight values = 10^{-6}
  - $\beta_C = 2.36*10^{-4}$
  - $\beta_M = 0.01$
  - Sample time = 0.01 s

Hardware: PID Control

- Control was performed independently on two gimbal axes
- We performed PID control in hardware with the following values
  - P = 0.0330
  - I = 0.0050
  - D = 0.330
  - Sample time = 0.0001 s
- PID Controller:

$$C(s) = P + \frac{I}{s} + Ds$$





Conclusion

- Initial simulations show that the ANC system is able to control the laser targeting system to follow a reference signal.
- Hardware experiments show that with a small disturbance, the control action of the ANC system causes the laser to significantly oscillate around the reference signal.
- A simple PID controller outperforms the ANC system.
- The sample time of the PID controller is 100 times faster than that of the ANC system.
- This sample time limitation is directly due to the processing capabilities of the dSpace control board.

Future Work

- May
  - Convert floating point arithmetic to fixed point via Matlab's Fixed Point Toolbox
  - Implement nonlinear neural function via lookup table
  - Implement division using fixed point arithmetic
- June
  - Build and simulate linear system replicator in Xilinx / System Generator
  - Compare simulation results to that of Matlab / Simulink version of linear system replicator
  - Build and simulate general system replicator in Xilinx / System Generator
  - Compare simulation results to that of Matlab / Simulink version of general system replicator

Future Work

- July
  - Build and simulate ANC system in Xilinx / System Generator
  - Compare simulation results to that of Matlab / Simulink version
  - Examine resiliency via simulation of both versions of the ANC system
- August
  - Implement linear system replicator, general system replicator, and controller in hardware via FPGA
  - Examine resiliency to the following anomalies: plant parameter changes, inter-system latencies, sensor data alteration, and false data injection
  - Compare control and resiliency results to a PID controller in hardware

References

- Hagan, M. and Demuth, H. (1999). Neural networks for control. In American Control Conference, 1999. Proceedings of the 1999, volume 3, pages 1642–1656 vol.3.
- Hyland, D. (1991). Neural network architecture for online system identification and adaptively optimized control. In Decision and Control, 1991., Proceedings of the 30th IEEE Conferenceon, pages 2552–2557 vol.3.
- Hyland, D. (1995). Adaptive neural control for flexible aerospace systems: progress and prospects. In Intelligent Control, 1995., Proceedings of the 1995 IEEE International Symposium on, pages 3–8.
- Lange, R. (2005). Design of a generic neural network fpga-implementation.
- Omondi, A. and Rajapakse, J. (2006). FPGA Implementations of Neural Networks. Springer.
- Rieger, C. (2010). Notional examples and benchmark aspects of a resilient control system. In Resilient Control Systems (ISRCS), 2010 3rd International Symposium on, pages 64–71.
- Salaheen, F. (2013). Modeling and control of gimbaled laser target system. Master's thesis, Temple University.
- Wei, D. and Ji, K. (2010). Resilient industrial control system (rics): Concepts, formulation, metrics, and insights. In Resilient Control Systems (ISRCS), 2010 3rd International Symposium on, pages 15–22.
- Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University, Cambridge, MA.

Acknowledgement

I would like to thank:

- Dr. Chang-Hee Won for his patience and support throughout this project
- My commmittee members: Dr. John Helferty and Dr. Dennis Silage
- Firdous Saleheen
- Timothy Boger
- CSNAP lab members

This material is based upon work supported in part by National Science Foundation Grant AIS ECCS-0969430 and REU Supplement.

Thank you.
Questions?