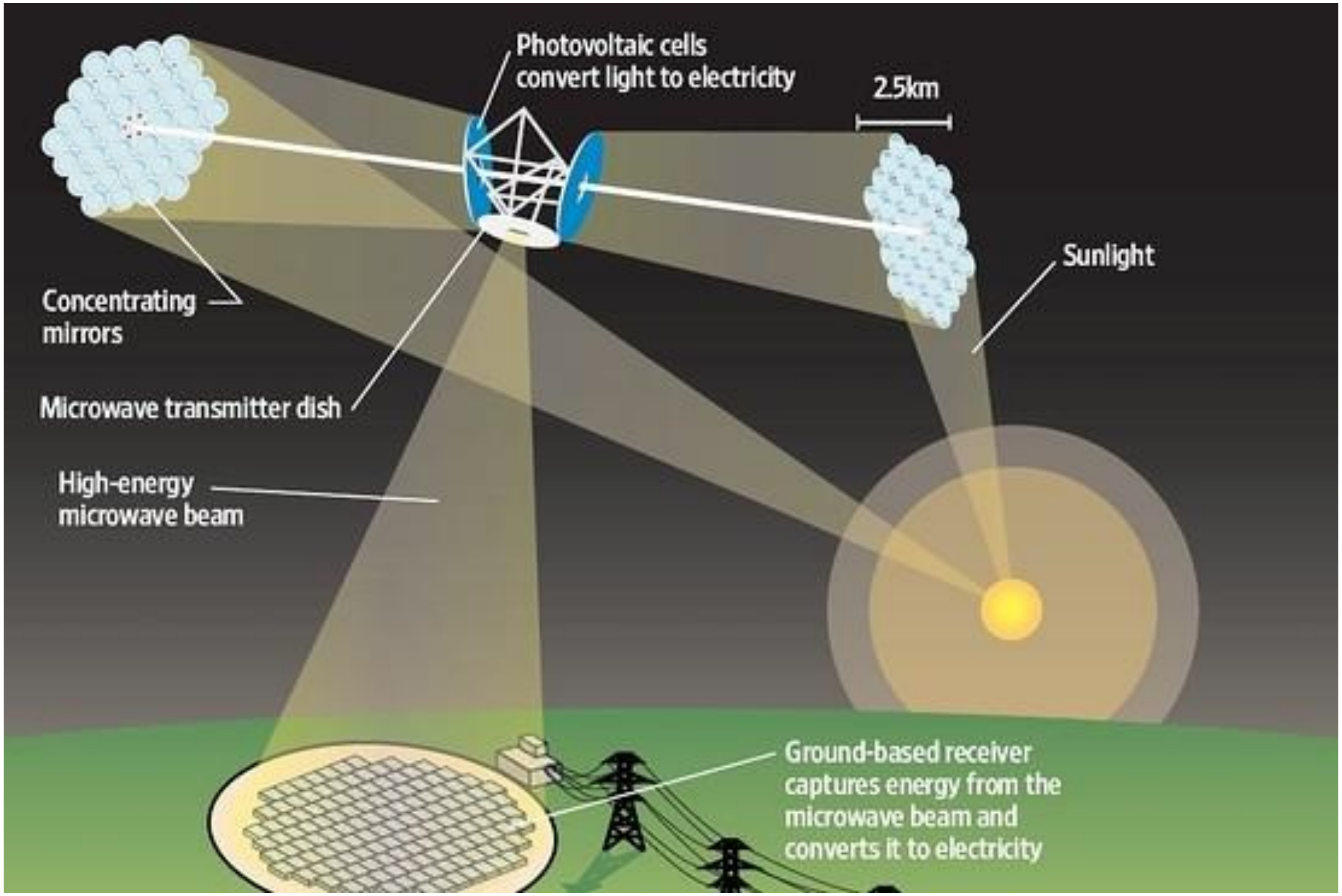
ADAPTIVE NEURAL CONTROL OF A GIMBALED LASER TARGETING SYSTEM

M.S. Thesis Defense

Salvatore Giorgi

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

Motivation

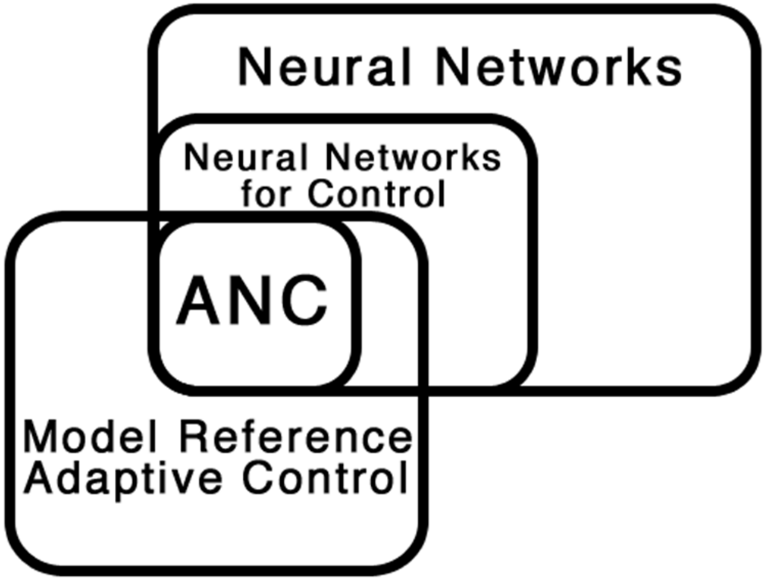

Photovoltaic cells
convert light to electricity
2.5km
Sunlight
Concentrating mirrors
Microwave transmitter dish
High-energy microwave beam
Ground-based receiver captures energy from the microwave beam and converts it to electricity

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Introduction

- Uses neural networks for system identification and control
- Plant's output is controlled to match an ideal reference system
- First proposed in the 1990's by D. C. Hyland
- First tested in hardware before analytical results were completed
- Resiliency of system has not been extensively studied
- Because of the computational complexity of the ANC system, proper hardware implementation presents a challenge



Objective

Investigate control capabilities of ANC System

- Control line of sight of laser targeting system as application
- Examine resiliency of ANC System
- Implement controller in both software and hardware

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Background

1997:
- Hyland: ANC system for ASTREX testbed at Airforce Philips Laboratory

1997
- Hyland: Convergence results for ANC system

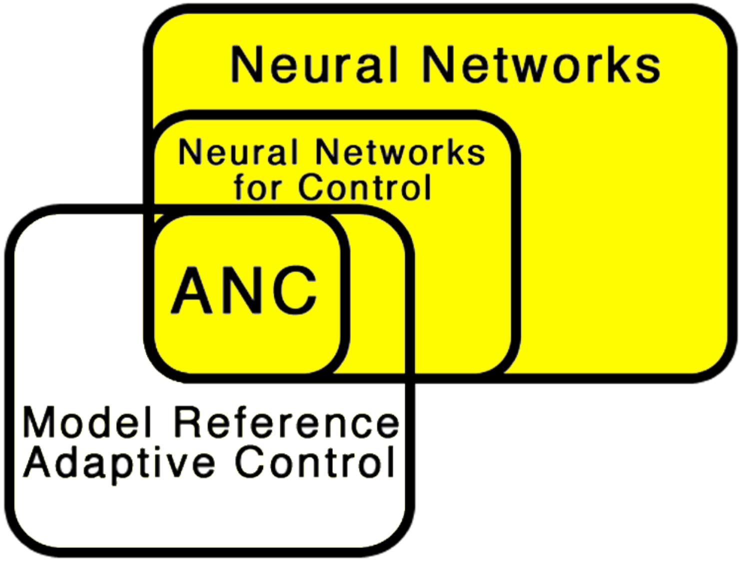
1998
- Lee, Ge and Wong: Passive line of sight stabilization, with 2 degree of freedom gimbal

2001
- Lin and Hsiao: Feed forward control of gimbaled antenna for disturbance rejection of guided missle

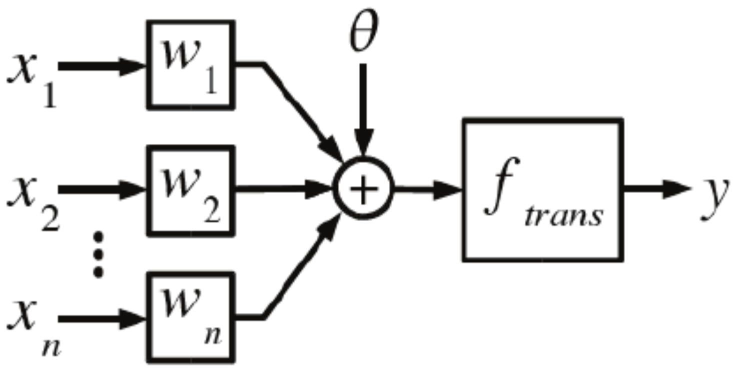
2008
- MacKunis, et all: Satellite attitude control via neural networks

Outline

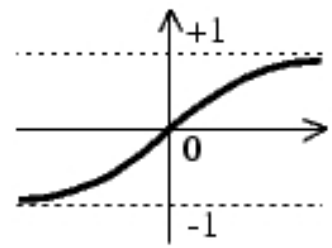
- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions



Neural Networks: Neurons / Neural Functions



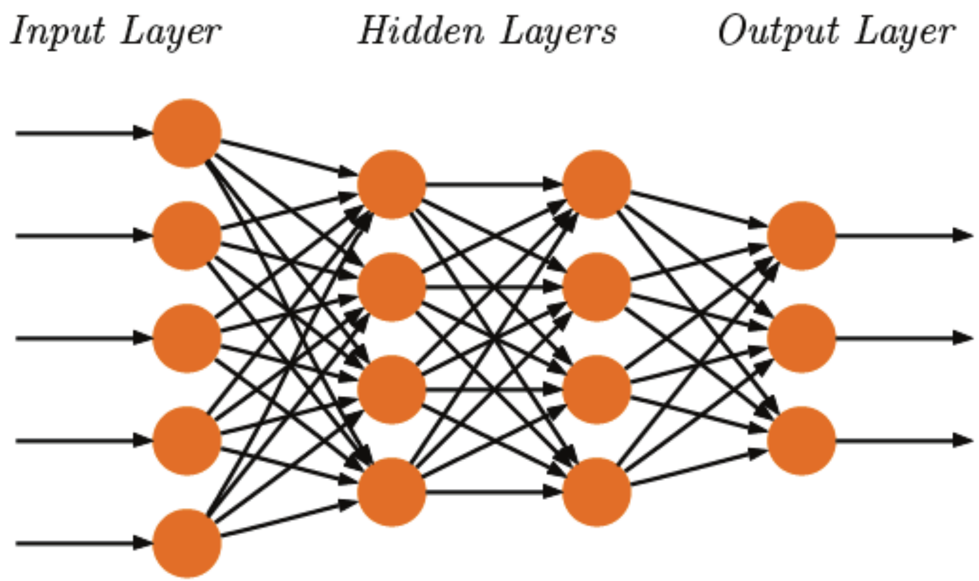
$$y = f_{trans}\left(\sum_{i=1}^{n} x_i w_i + \theta\right)$$



- 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

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

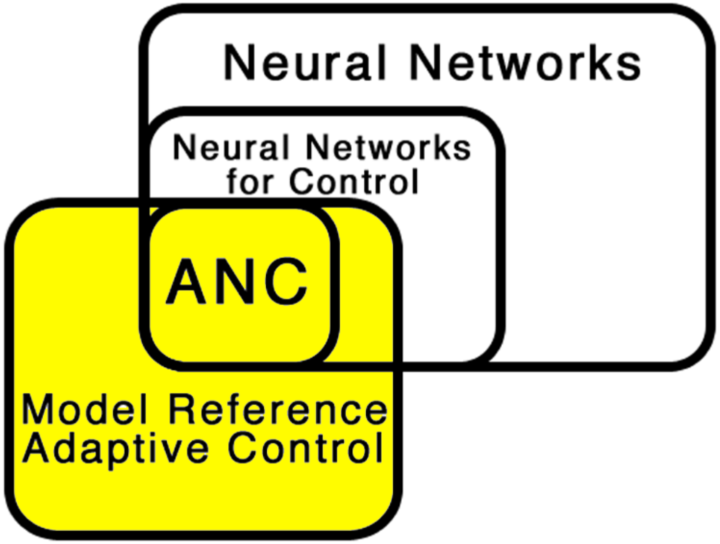
Neural Networks: Layers



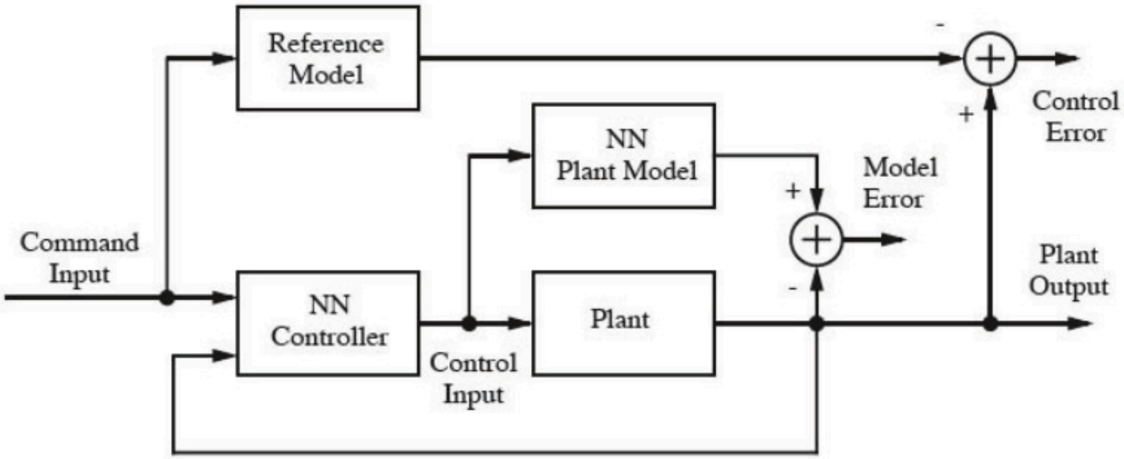
- Simplest neural networks consist of an input and output layer.
- Most neural networks contain at least one hidden layer.
- The ANC system used consists of a linear input layer, a single nonlinear hidden layer, and a linear output layer.

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

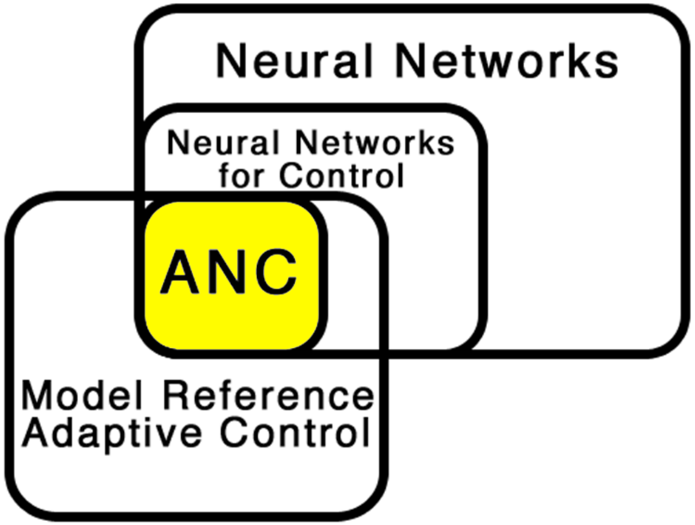


Model Reference Adaptive Control



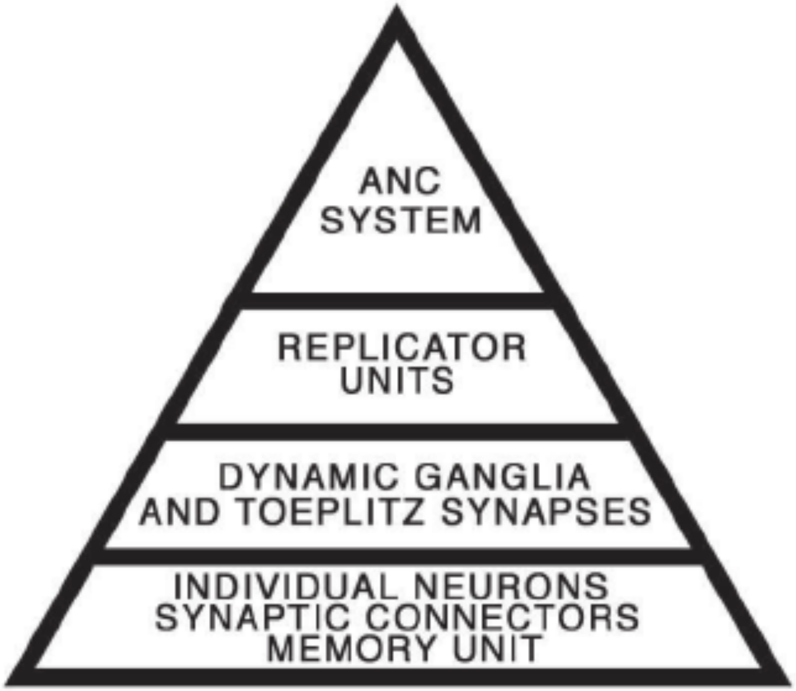
- Model Reference Adaptive Control (MRAC): Desirable dynamic characteristics of the plant are specified in a reference model
- Input / adaptable plant parameters are changed so that the plant's output matches the reference's output
- Two independent neural networks are used:
  - One replicates the plant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

Outline

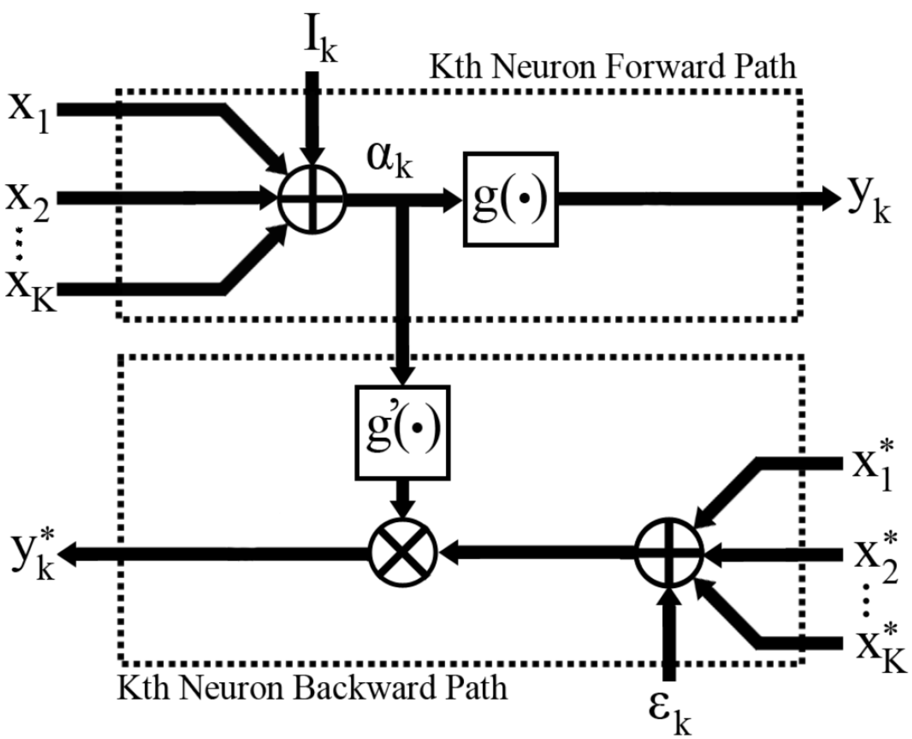


- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

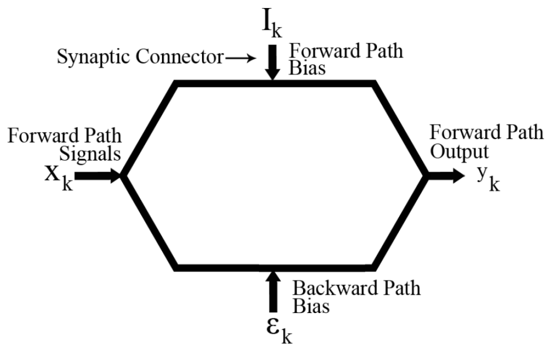
ANC System: Hierarchy



ANC System: Individual Neuron

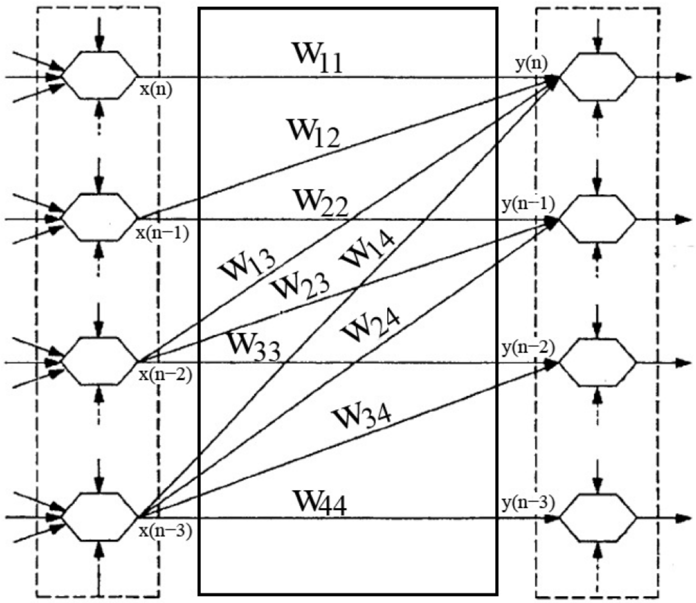


- Neurons are bi-directional devices with a forward path and a backward path
- Each neuron contains a neural function, which is either a linear or a sigmoid function
- Derivative of neural function is multiplied by sum of backward path signals

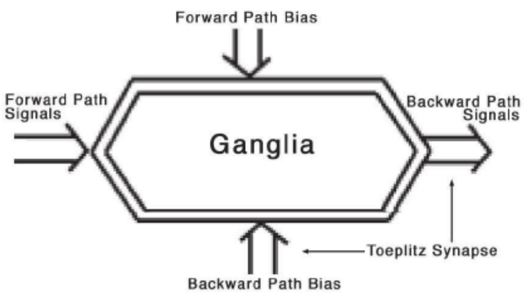


ANC System: Dynamic Ganglia

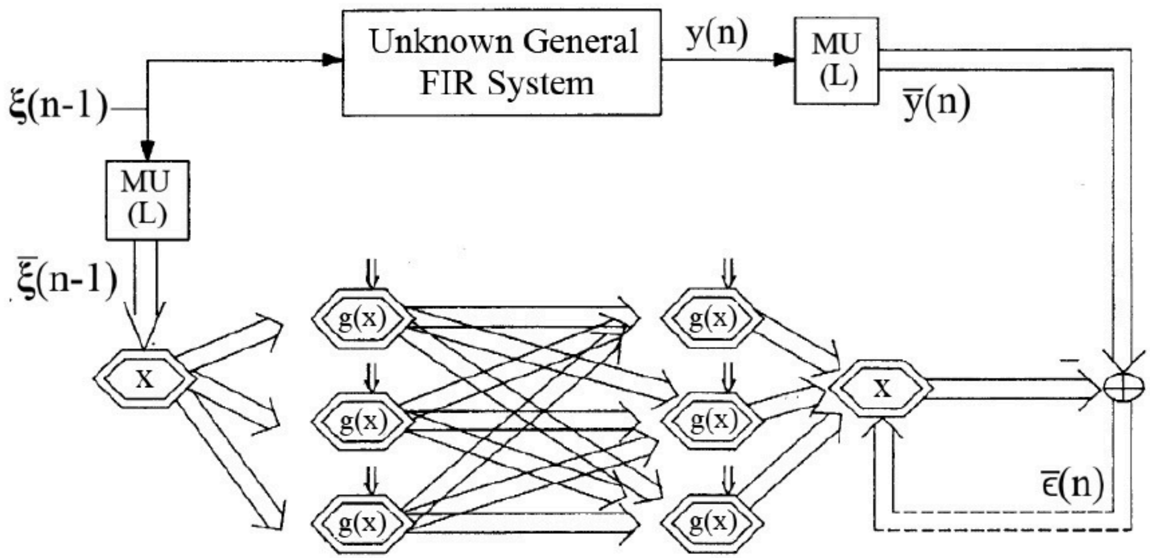
- The position of the neurons determine the age of the data
  - Top level neurons represent current data
  - Lower level neurons represent past data
- Past data points do not depend on future inputs



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



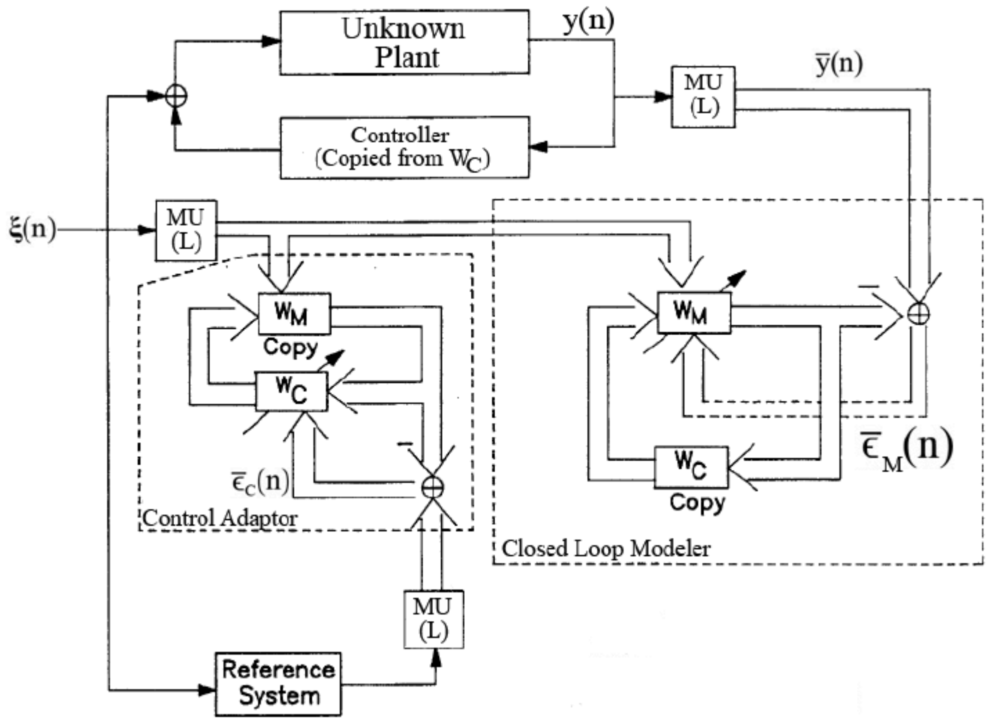
ANC System: FIR Replicator Unit



- 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: 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



Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Resilient Control: Definition

- **Resiliency** is defined as the capacity of a 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

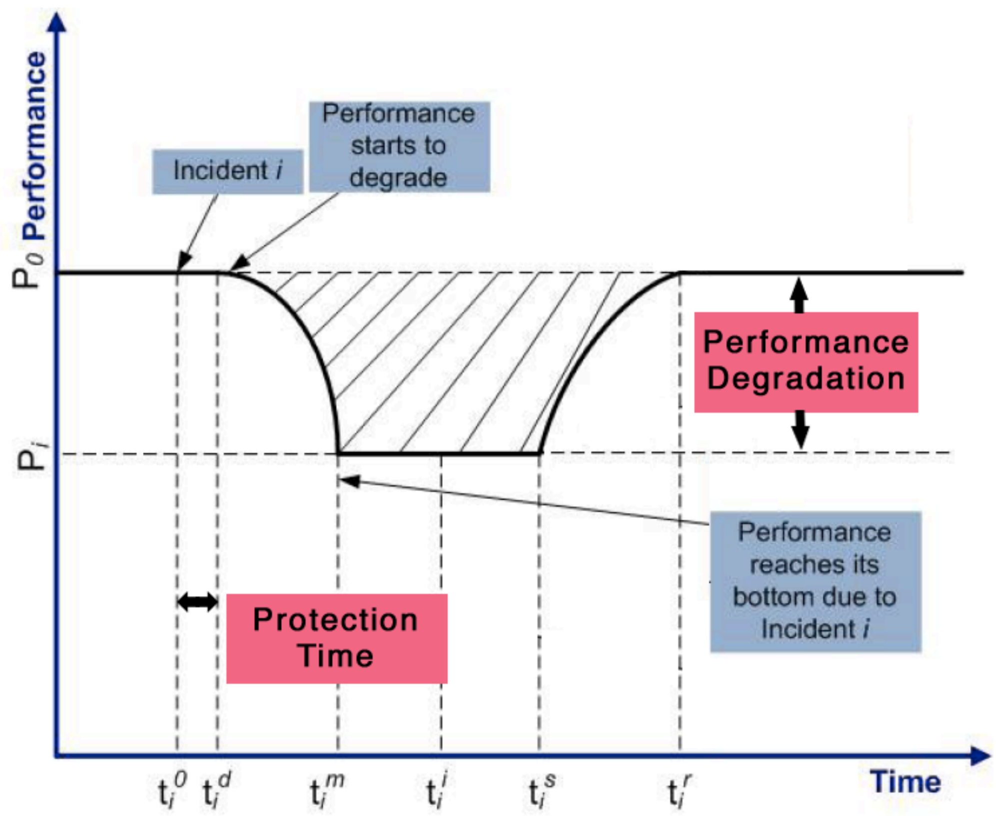
Resilient Control: Metrics

- **Performance Degradation**
  maximal performance degradation due to incident *i*

$$P_i^d = P_0 - P_i$$

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

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



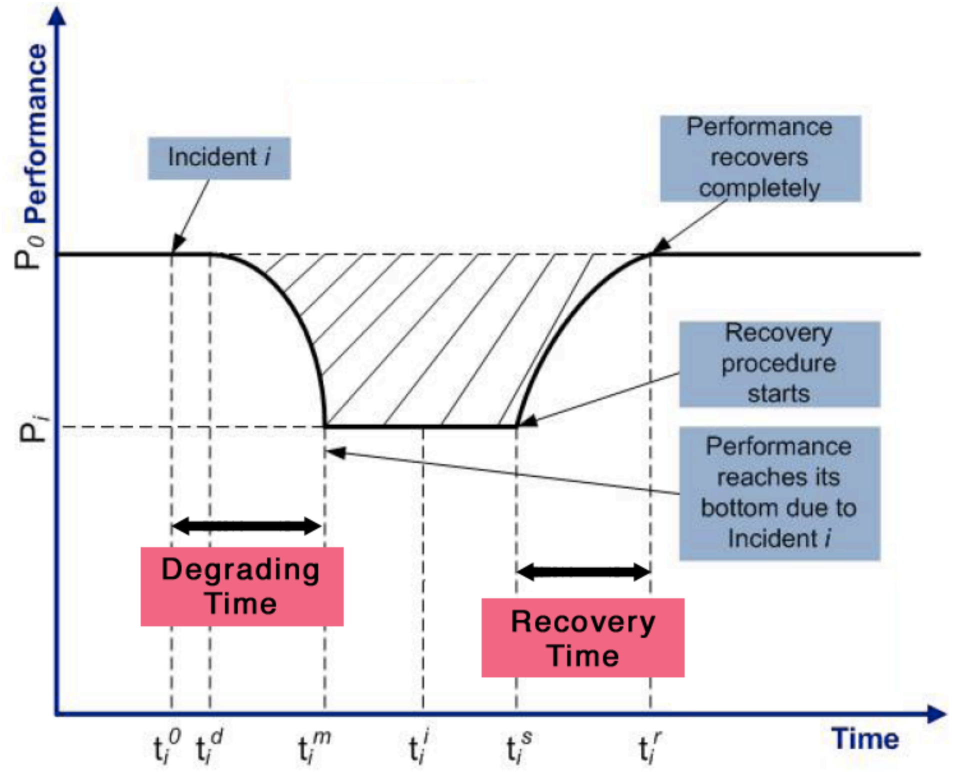
Resilient Control: Metrics

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

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

- **Recovery Time**
  the time that the system needs to recover to normal operation from the incident *i*

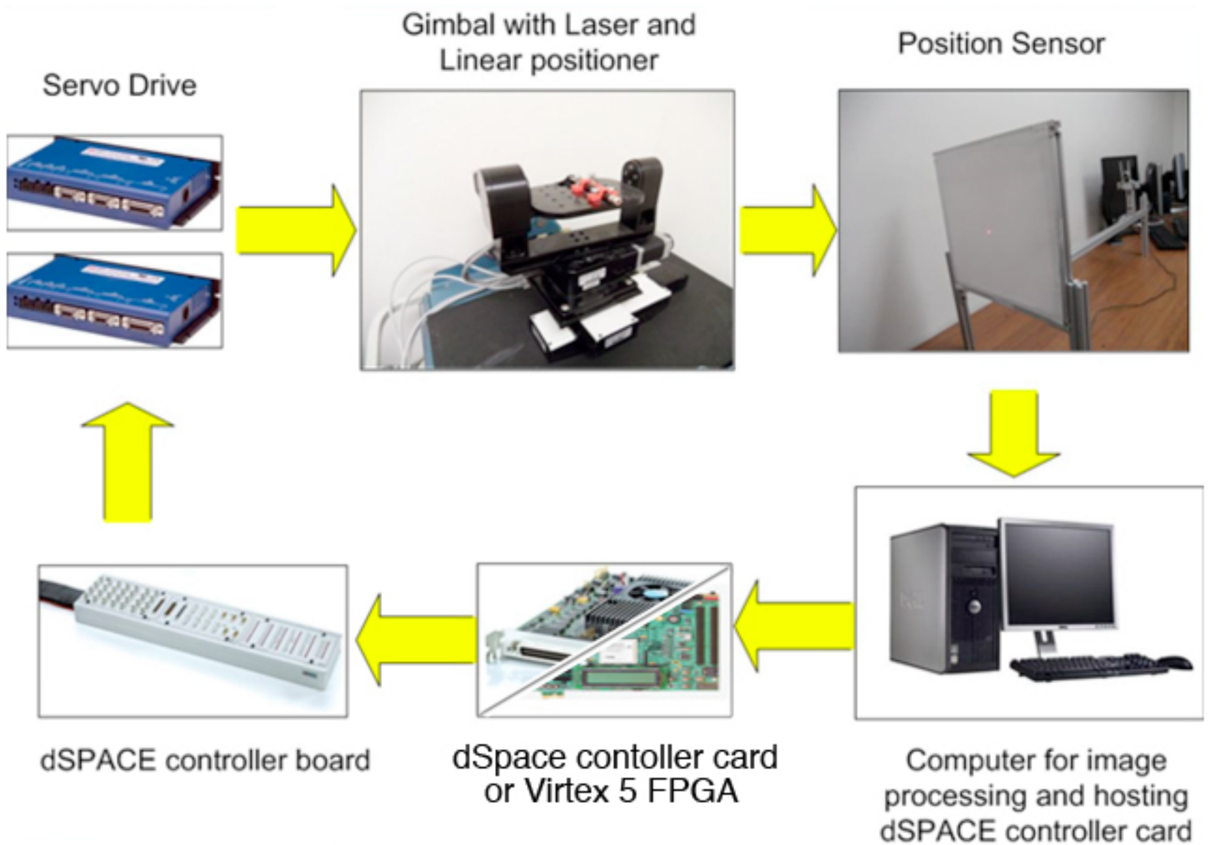
$$T_i^r = t_i^r - t_i^s$$



Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Laser Targeting System: Test Bench



This test bench was created by Timothy Boger, Sudarshan Kandi, Ross Keyes.

Laser Targeting System: Plant Model

Yaw Axis State Space Equations:

$$dx_{AZ}(t) = \left(A_{AZ}x_{AZ}(t) + B_{AZ}u_{AZ}(t)\right) + G_{AZ}dw_{AZ}(t)$$

$$y_{AZ}(t) = C_{AZ}x_{AZ}(t) + v_{AZ}(t)$$

State and Control Vectors:

$$x_{AZ}(t) = \left[f_y(t) \quad \omega_a(t) \quad i_{aAZ}(t) \quad i^*_{iAZ}(t) \right]^T$$

$$u_{AZ}(t) = f^*_x(t)$$

$$A_{AZ} = \begin{bmatrix} 0 & C_x & 0 & 0 \\ -\dfrac{K_{a\omega}}{C_x J_a} & -\dfrac{K_{af}}{J_a} & \dfrac{k_b N}{J_a} & 0 \\ 0 & 0 & -\dfrac{1}{\tau_{iAZ}} & \dfrac{K_{iAZ}}{\tau_{iAZ}} \\ 0 & 0 & 0 & -\dfrac{1}{\tau_{sAZ}} \end{bmatrix}$$

$$B_{AZ} = \begin{bmatrix} 0 & 0 & 0 & \dfrac{K_{sAZ}C_{\omega AZ}}{\tau_{sAZ}} \end{bmatrix}^T$$

$$G_{AZ} = \begin{bmatrix} 0 & J_a^{-1} & 0 & 0 \end{bmatrix}^T$$

$$C_{AZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Assumptions

- Negligible coupling effect
- Gimbal sits on fixed platform
- Line of sight angle is small
- Neglecting high order nonlinear terms

Model Developed by Firdous Saleheen

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Results: Control of GLTS

Control Performance Metrics

- Reference/Pointing Error

$$\varepsilon(t) = x_{ref}(t) - x_{act}(t)$$

- Maximum Absolute Value

$$\max_t \{|x_{act}(t)|\}$$

- Root Mean Square Reference Error (RMSE)

$$\mu_\varepsilon = \sqrt{\frac{\sum_{t=t_0}^{t_f} \varepsilon^2(t)}{N_\varepsilon}}$$

- Standard Deviation of Reference/Pointing Error (SDRE/SDPE)

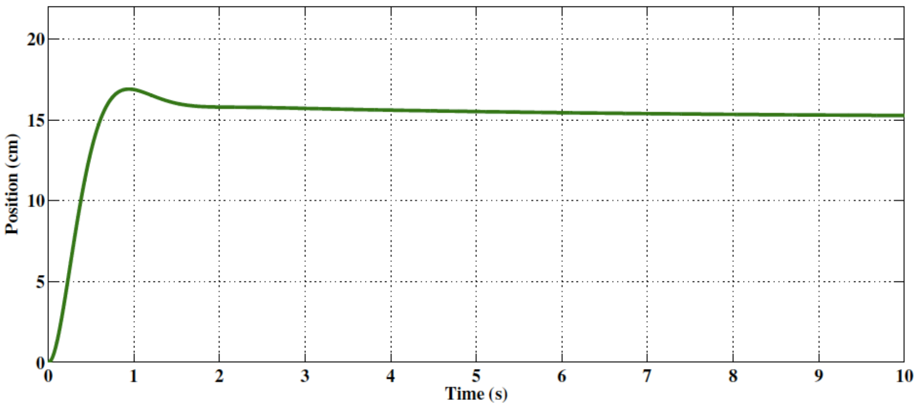
$$\sigma_\varepsilon = \sqrt{\frac{\sum_{t=t_0}^{t_f} (\varepsilon(t) - \mu_\varepsilon)^2}{N_\varepsilon - 1}}$$

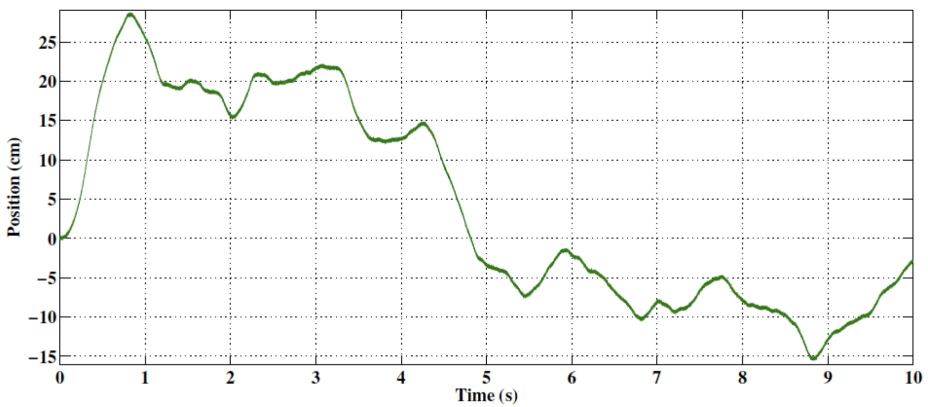
Results: Control of GLTS

PID Control

- We control the line of sight of the GLTS to match 15.15 cm.
- Both measurement and process noise are added the system with values $W_m = 10^{-6}$ and $W_p = 10^{-3}$.
- The PID Controller had the following form, with values $K_p = 1.04\text{x}10^{-2}$, $K_I = 2.43\text{x}10^{-4}$, $K_D = 4.73\text{x}10^{-2}$, $N = 6.45$:

$$C(s) = K_P + K_I\frac{1}{s} + K_D\frac{Ns}{s+N}$$





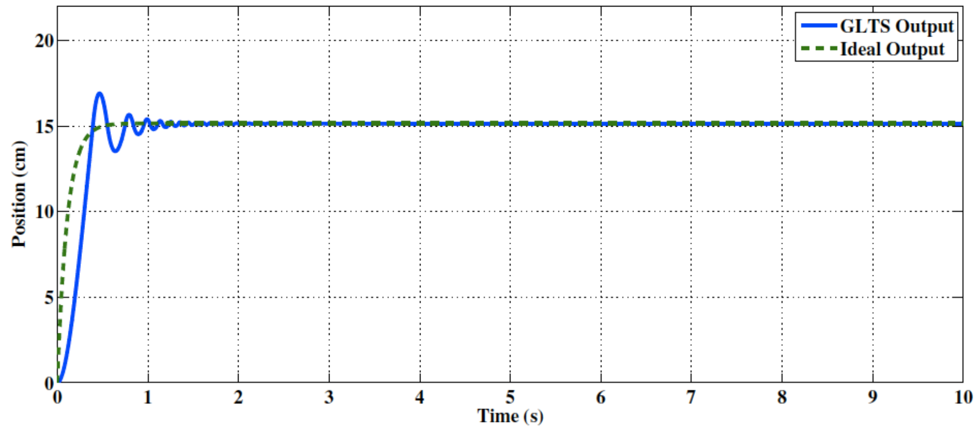
Results: Control of GLTS

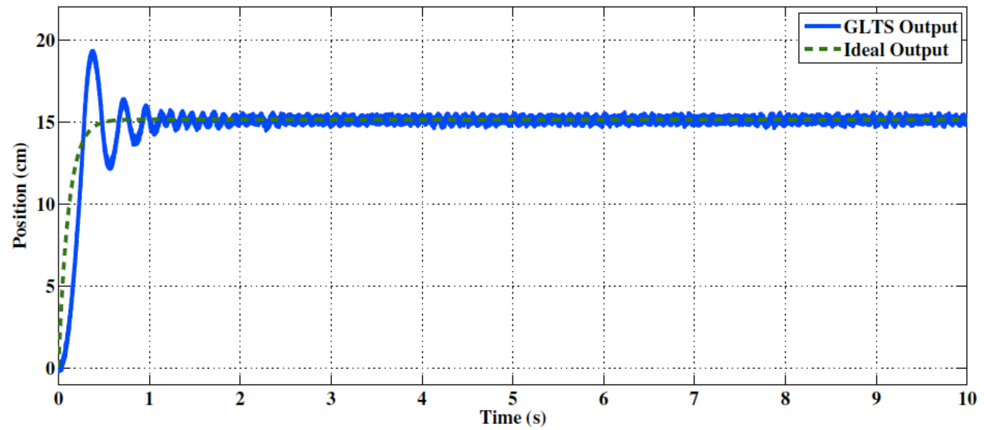
ANC System Control

- We control the GLTS with the ANC System
- Same simulation values as PID control simulation
- The following reference system was used

$$T_I\ (s) = \frac{9}{s+9}$$

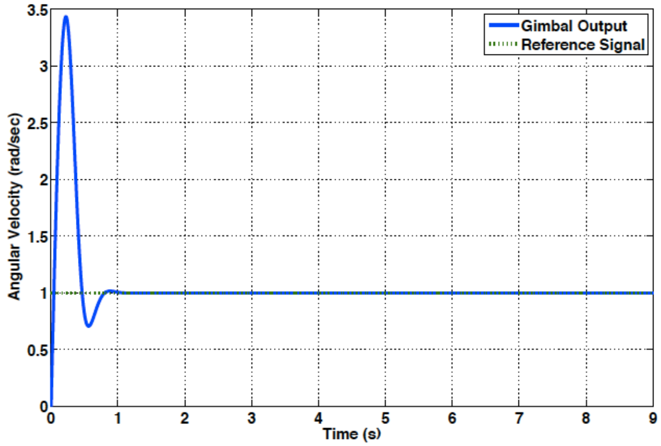
Controller	Maximum Value (cm)	RMSE	SDPE
PID	36.32	17.56	32.36
ANC	19.30	0.15	0.24



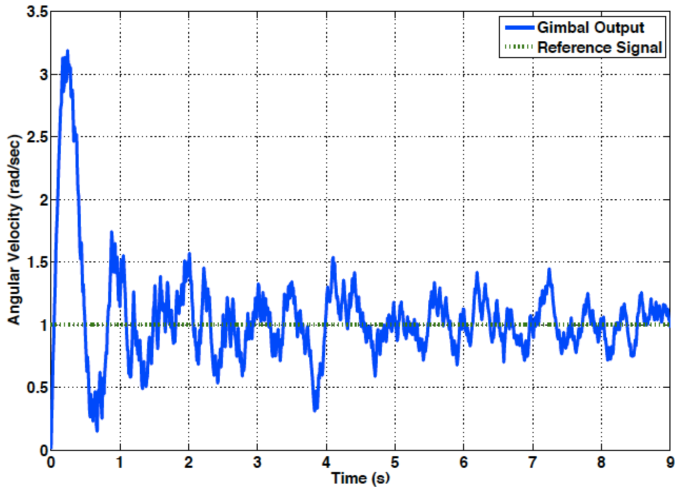


Results: Control of Nonlinear Gimbal

- We control a nonlinear two-axis gimbal model.
- Instead of controlling line of sight, our objective is to stabilize the angular velocity in the presence of noise.
- Non-zero signal must be used for ANC, so we stabilize around 1 rad/sec



Maximum Value (rad/sec)	RMSE	SDPE
3.19	0.19	0.28



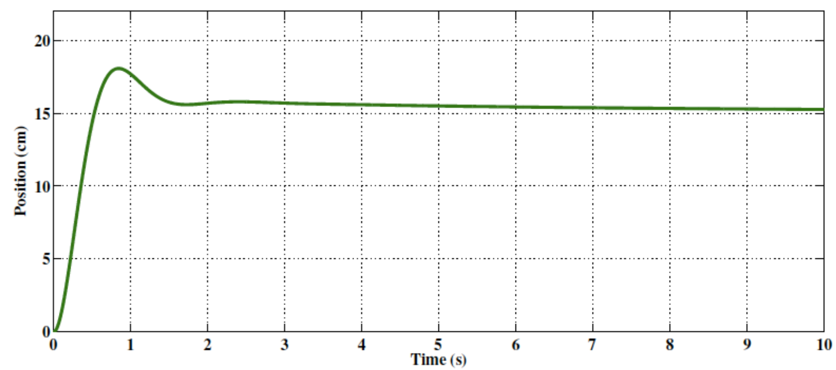
Outline

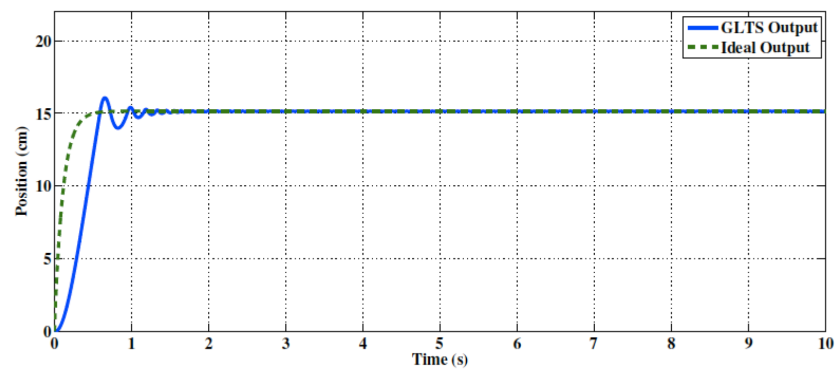
- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Results: Resiliency of ANC System

Added Latency Attack

- We use the same simulation values as the control simulations above.
- The attack occurs at t = 0.
- We add a 25 sample delay within the feedback loop.
- We see that both the PID (top figure) and ANC System (bottom figure) are able to maintain control.

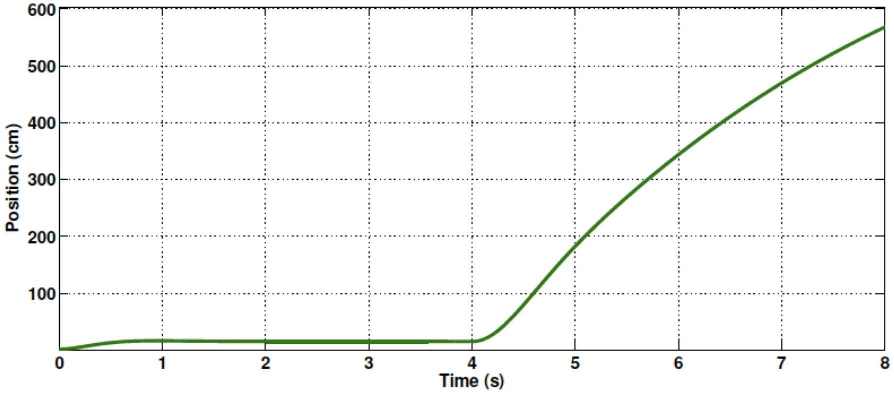


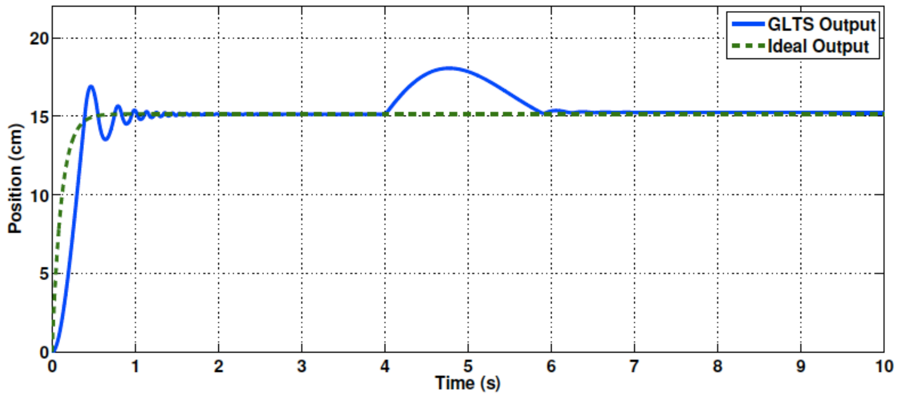


Results: Resiliency of ANC System

False Data Injection Attack

- We add a constant input to the controller in the form of 10 cm.
- This attack occurs at t = 4.
- We see that while the ANC System is able to maintain control within 2 seconds, the PID controller is never able to recover.

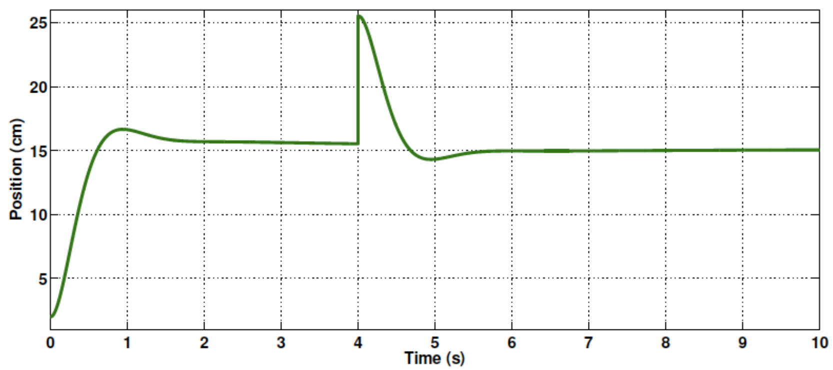


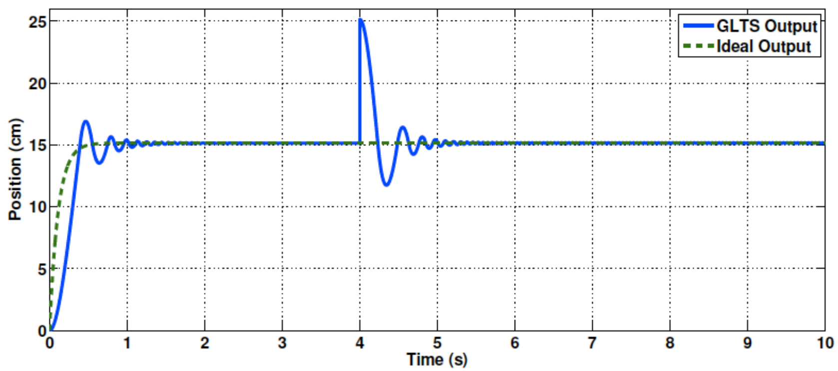


Results: Resiliency of ANC System

Sensor Data Alteration Attack

- We add a value of 25 cm to the output of the plant.
- This attack occurs at t = 4.
- Both controllers are able to maintain control, with similar initial overshoot.
- The ANC System shows some oscillation, but both controllers return to operational normalcy within 1-2 seconds.

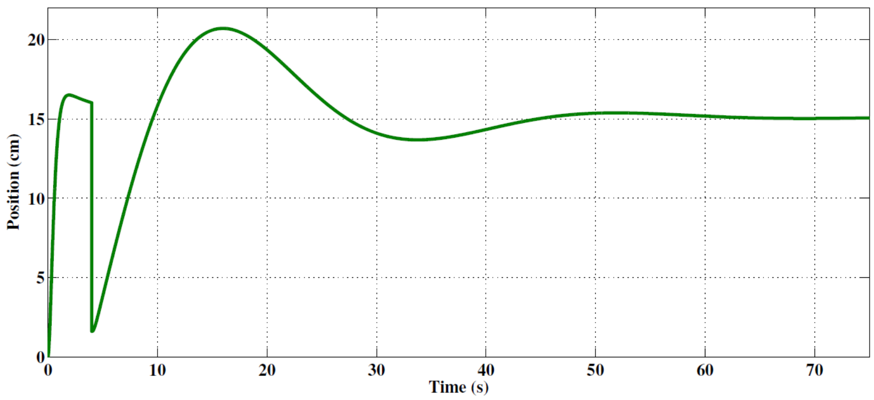


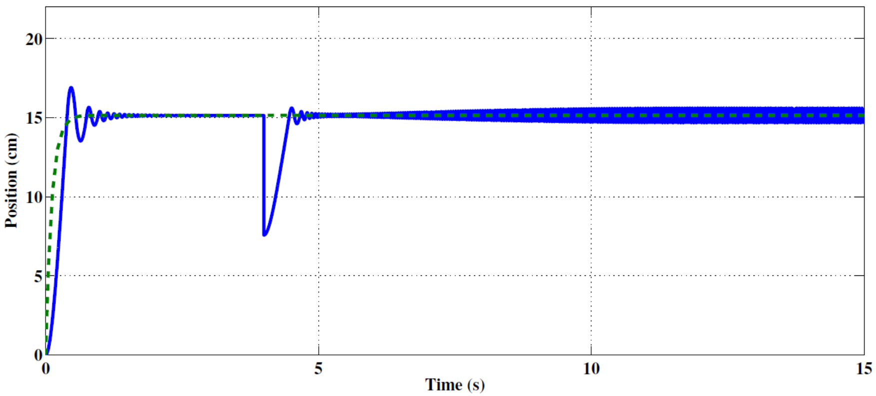


Results: Resiliency of ANC System

Plant Parameter Change Attack

- We change the parameters:
  - B_{AZ} => $10B_{AZ}$
  - C_{AZ} => C_{AZ} / 10
- This attack occurs at t = 4.
- The PID controller has a larger undershoot than the ANC System.
- The PID controller recovers after approximately 55 seconds.
- The ANC System reestablishes the proper line of sight, but with growing oscillations.





Results: Resiliency of ANC System

Resilient Metrics

- The following table summarizes the results of the attacks, using the resilient metrics defined above

Metric	Added Latency		False Data Injection		Sensor Data Alteration		Plant Parameter Changes	
	PID	ANC	PID	ANC	PID	ANC	PID	ANC
T_i^r	16.84 s	0.65 s	∞	1.5 s	20 s	1 s	70 s	∞
P_i^d	-2.92 cm	-0.92 cm	∞	-2.85 cm	-10.45 cm	-9.95 cm	13.55 cm	7.58 cm
T_i^p	0 s	0 s	0 s	0 s	0 s	0 s	0 s	0 s
T_i^d	0.86 s	0.65 s	∞	0.7 s	0 s	0 s	0 s	0 s

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Results: ANC Using FPGA

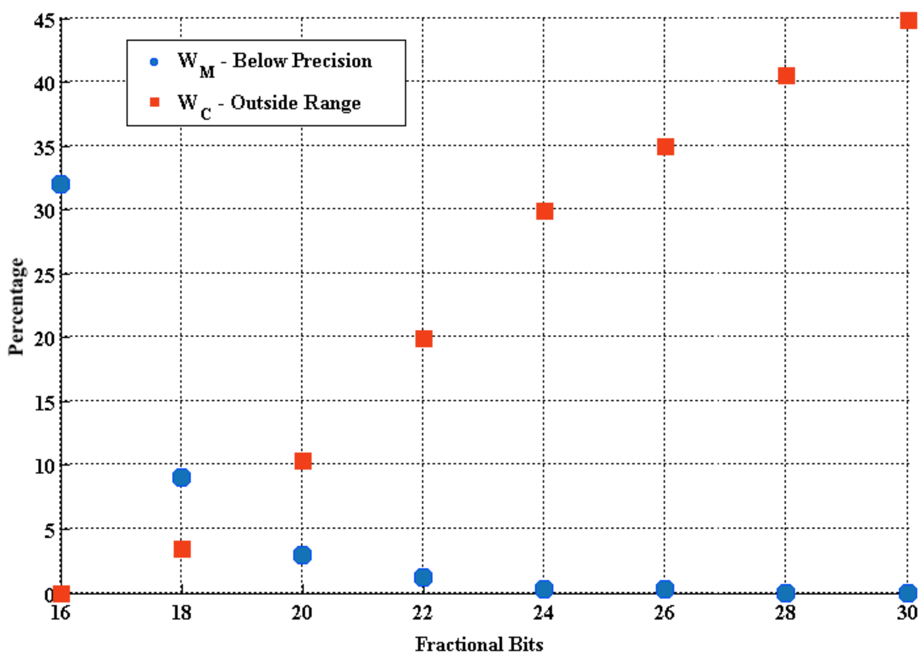
Simulations, “Hardware in the Loop”, and Experiments

- FPGA simulations are done with the Xilinx block set within Simulink
  - The simulations show exactly how our model will behave when implemented in hardware.
- “Hardware in the Loop” simulations run the model on the FPGA hardware within the simulation.
  - Values are sent from Simulink to the FPGA and back to Simulink.
  - This is often done to speed up the simulation process.
- Experiments are done with our model running on a processor (dSPACE or FPGA) with the GLTS test bed.
  - dSPACE experiments convert Simulink diagrams to C code, which then runs on the dSPACE board in real time.

Results: ANC Using FPGA

Floating Point to Fixed Point Conversion

- We ran a control simulation, and gathered values of all of the weights
- Gathered values by ganglia location within ANC System
- Used NumericTypeScope to convert floating point data



- **Outside Range**
  - data cannot be represented
- **Below Precision**
  - data is underspecified

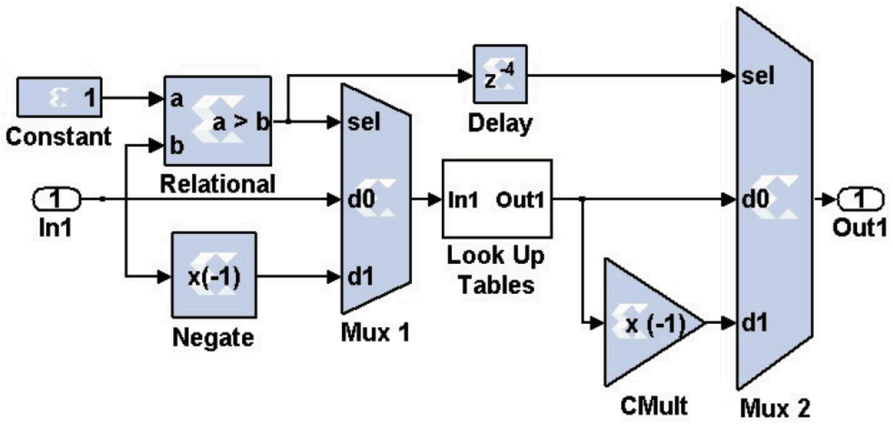
Results: ANC Using FPGA

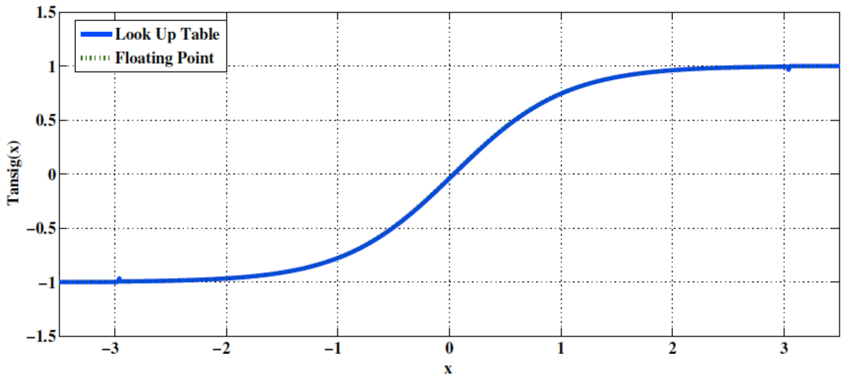
Nonlinear Neural Function

- Tansig function is of the form

$$f(x) = \frac{2}{1+e^{-2x}} - 1$$

- We take advantage of tansig being an odd function f(-x) = -f(x)
- Also, f(x) = 1 for x >|3|
- Use a look up table for f(x) with x in [0,3]
- We have small discrepancy when x = -3,3, which we neglect.





Results: ANC Using FPGA

Simulation: System Replicator

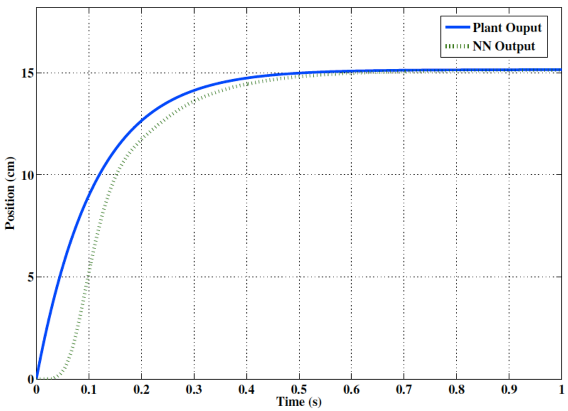


Fig: Linear Replication

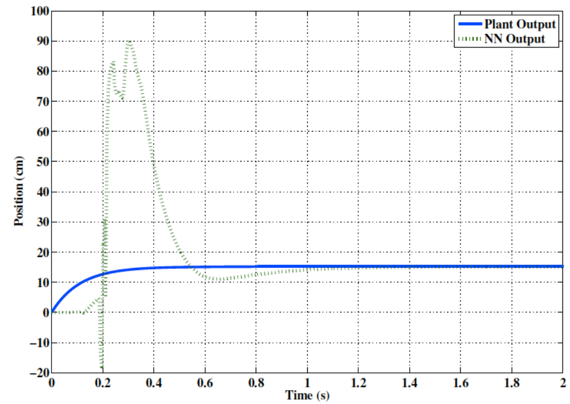


Fig: Nonlinear Replication

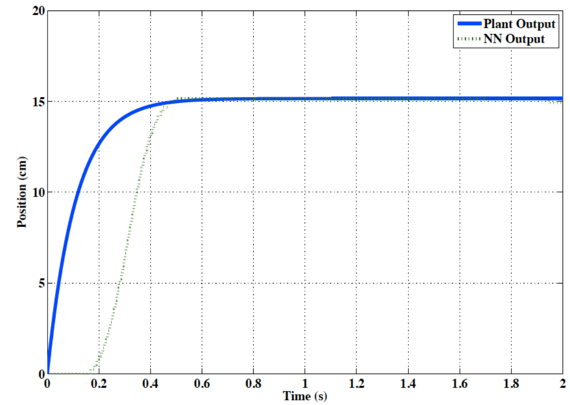


Fig: Multiplexed Multipliers

We replicate the following system with three different Replication Unit Architectures

$$T_I\ (s) = \frac{9}{s+9}$$

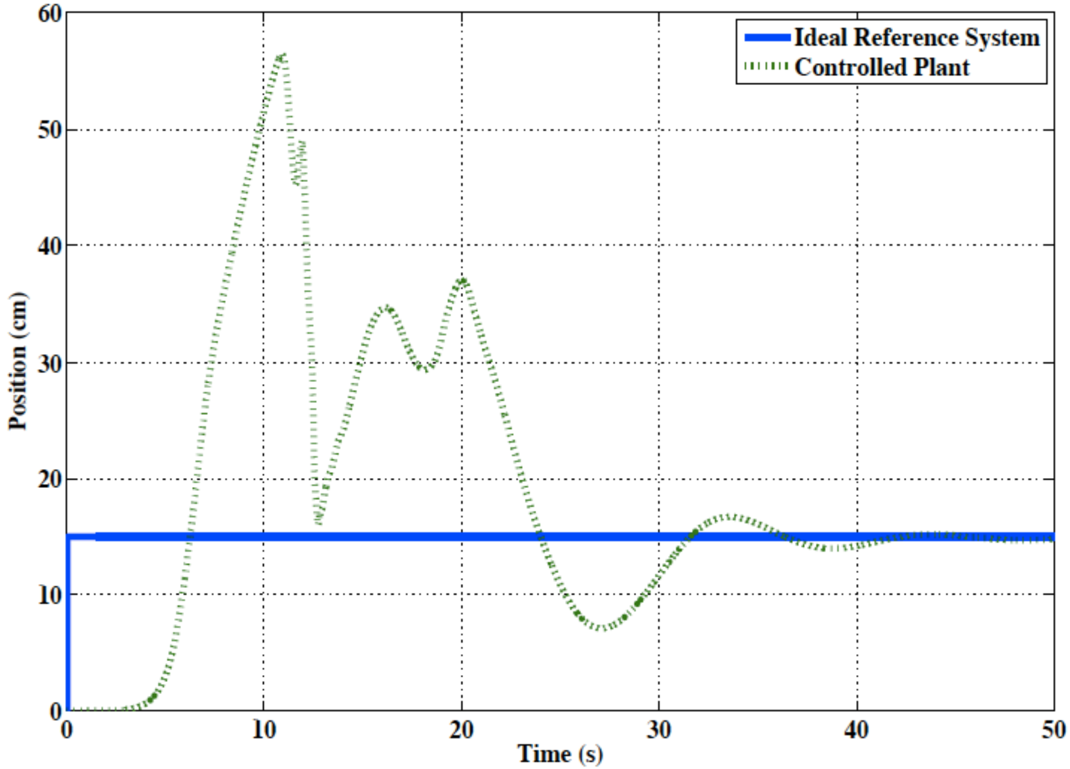
Results: ANC Using FPGA

Simulation: ANC System

- We control the output of the linear GLTS model to follow the output of

$$T_I\ (s) = \frac{9}{s+9}$$

- Input is a step signal with 15.15 amplitude applied at t=0
- Each Replicator Unit has one hidden layer with three nonlinear ganglia, 3 neurons per ganglia



Results: ANC Using FPGA

Hardware Implementation: Data Transfer, Clock Mode, and Shared Memory

- In "hardware in the loop" simulation, the FPGA clock is in sync with the Simulink clock.
- For real time control, we would like to use the fastest clock speed possible on the FPGA.
- The Simulink clock is constrained by the image processing time.
- In order to have two separate clocks, we use shared memory between the PC and FPGA, via two FIFO pairs (one input FIFO / output FIFO per pair).
- The memory on the PC and the memory on the FPGA now share the same address.
- When image processing data is ready, it is immediately written to FPGA via a ethernet connection through shared FIFO.
- Similarly, when control signal is ready on FPGA, it is immediately written to PC.

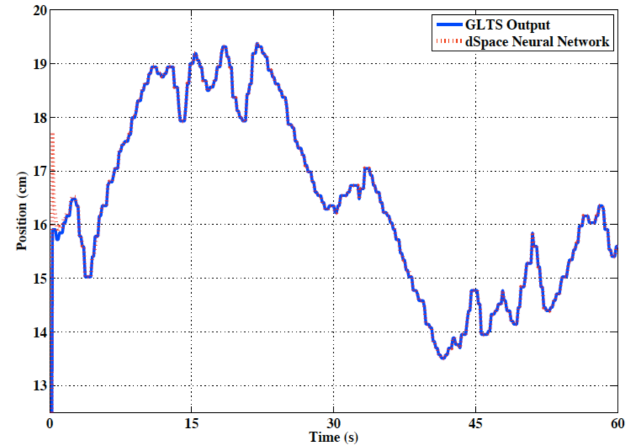
Results: ANC Using FPGA

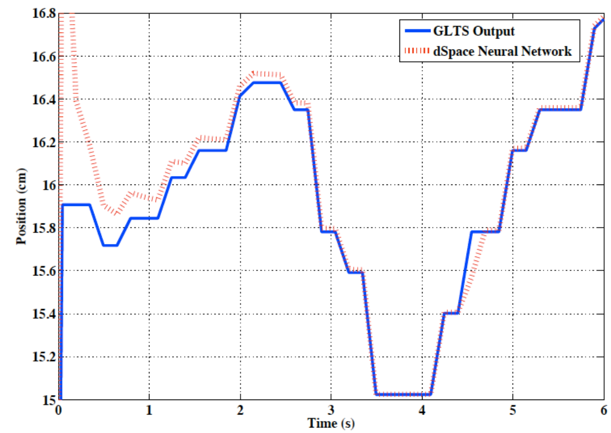
Hardware Implementation:

System Replication with dSPACE

- For experiment, we consider maximum value and RMSE as well as percent overshoot.
- We use a linear FIR Replicator Unit with 3 neurons per ganglia.
- We define percent overshoot as:

$$PO = \frac{\text{max value} - \text{step value}}{\text{step value}}$$





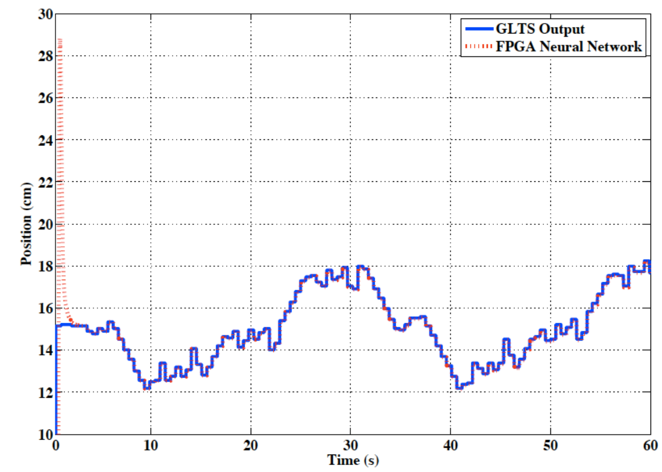
Results: ANC Using FPGA

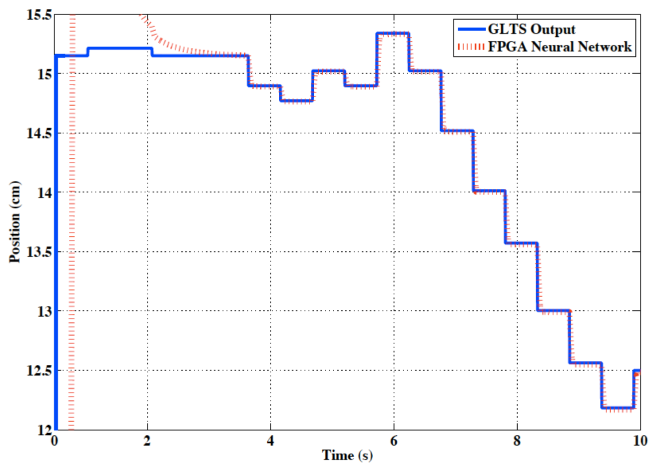
Hardware Implementation:

System Replication with FPGA

- Same parameter values and Replicator Unit architecture as dSPACE experiment.
- Here we used the multiplexed multiplier architecture, with multipliers optimized for speed.
- The results of the experiment are shown in the following table:

Hardware	Max Value	Step Value	Percent Overshoot	RMSE
dSPACE	17.72	15.91	0.11	0.035
FPGA	28.80	15.15	0.90	0.030





Results: ANC Using FPGA

Hardware Implementation: Hardware Resources

- We monitored the hardware resources for various Replicator Unit designs (including input/output FIFOs) using different clock rates and FPGAs.
- Multipliers were optimized for either Area or Speed.
- The table below shows the results of a 10 neurons per ganglia linear FIR replicator unit, on the Virtex 6 FPGA running at 100 MHz.

Optimization	Hardware	Percentage Used
Area	Slice Registers	157%
	Slice LUTs	594%
	DSP48e Slices	110%
Speed	Slice Registers	51%
	Slice LUTs	39%
	DSP48e Slices	420%

Results: ANC Using FPGA

Hardware Implementation: Hardware Resources

- This table shows the hardware resources of a linear FIR System Replicator, with 3 neurons per ganglia, using multiplexed multipliers, on the Virtex 6 FPGA

Clock (MHz)	Optimization	Hardware	Percentage Used
100	Area	Slice Registers	31%
		Slice LUTs	44%
		DSP48e Slices	12%
	Speed	Slice Registers	23%
		Slice LUTs	12%
		DSP48e Slices	27%
50	Area	Slice Registers	30%
		Slice LUTs	39%
		DSP48e Slices	13%
	Speed	Slice Registers	23%
		Slice LUTs	12%
		DSP48e Slices	27%

Outline

- Motivation
- Introduction and Objective
- Background
  - Neural Networks
  - Model Reference Adaptive Control
- Adaptive Neural Control (ANC)
- Resilient Control
- Gimbaled Laser Targeting System (GLTS)
- Results
  - Control of GLTS and Nonlinear Gimbal Model
  - Resiliency of ANC System
  - ANC Using FPGA
- Conclusions

Conclusions: Control Simulations

- We see that the ANC System outperforms the PID controller in terms of maximum value, RMSE, and SDPE.

Controller	Maximum Value (cm)	RMSE	SDPE
PID	36.32	17.56	32.36
ANC	19.30	0.15	0.24

- The ANC System can control the nonlinear gimbal model to match a desired reference signal, in the presence of measurement and process noise.

Maximum Value (rad/sec)	RMSE	SDPE
3.19	0.19	0.28

Conclusions: Resiliency

- In terms of protection time T^p, both the PID and ANC System give the same results for each attack.
- For the Plant Parameter Change attack, PID outperforms the ANC System in terms of recovery time T^r, whereas the ANC System outperforms the PID in terms of performance degradation P^d.
- For all other attacks, the ANC System out performs or performs as well as the PID Controller in terms of each metric.

Metric	Added Latency		False Data Injection		Sensor Data Alteration		Plant Parameter Changes	
	PID	ANC	PID	ANC	PID	ANC	PID	ANC
T_i^r	16.84 s	0.65 s	∞	1.5 s	20 s	1 s	70 s	∞
P_i^d	-2.92 cm	-0.92 cm	∞	-2.85 cm	-10.45 cm	-9.95 cm	13.55 cm	7.58 cm
T_i^p	0 s	0 s	0 s	0 s	0 s	0 s	0 s	0 s
T_i^d	0.86 s	0.65 s	∞	0.7 s	0 s	0 s	0 s	0 s

Conclusions: Hardware Implementation

- Despite the larger initial overshoot for the FPGA implementation, both hardware implementations perform approximately the same in terms of RMSE.

Hardware	Max Value	Step Value	Percent Overshoot	RMSE
dSPACE	17.72	15.91	0.11	0.035
FPGA	28.80	15.15	0.90	0.030

- With 10 neurons per ganglia, we begin to max out hardware resources.
- Implementation of full ANC System not possible with current hardware.

Proposal vs Defense

Proposed Work	Completed Work
Build Software model of ANC System in Simulink	Complete: System Replication and Control simulations were successful
Simulate Control of GLTS in Simulink	Complete: Control simulations within Simulink were successful for linear GLTS model and nonlinear gimbal model
Build hardware model of ANC System in Xilinx/System Generator	Complete: Each subsystem was verified to behave properly
Simulate Control of GLTS with Xilinx/System Generator	Complete: Hardware model was able to control linear GLTS model
Use FPGA for real time replication and control of GLTS	Partially Complete: Real time system replication was achieved, but hardware limitations impeded full implementation of ANC System
Examine Resiliency of ANC System	Complete: Four separate attacks simulated and resiliency measured through resilient metrics

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

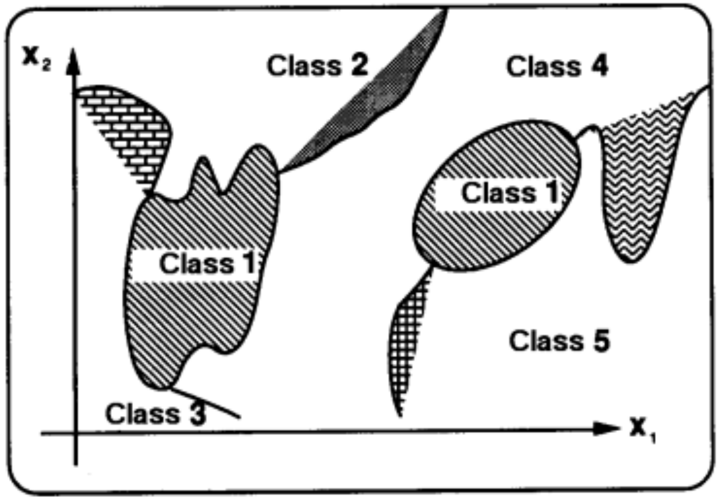
I would like to thank:

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

This material is based upon work supported in part by National Science Foundation Grant AIS ECCS-0969430 and REU Supplement, as well as the College of Engineering.

Thank you.
Questions?

Neural Networks: 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

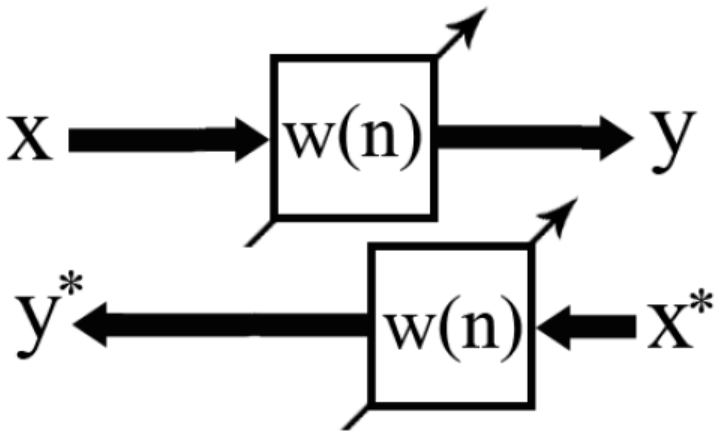
ANC System: Memory Unit

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

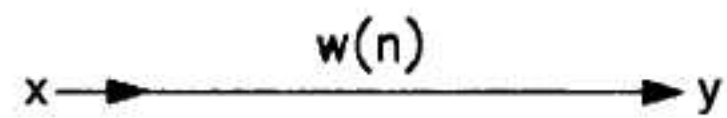
- Takes a scalar time-series input
- Produces an *L*-dimensional vector consisting of
  - Current input
  - *L*-1 delayed values
- Bar notation denotes a column vector with *L* past signal values

ANC System: 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: Weight Update Law

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

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

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

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

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

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

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

Resiliency vs Robustness / Fault-Tolerance / Adaptiveness

- **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
- Resiliency can be considered the superset of all of the above properties

Results: ANC Using FPGA

Division

- Instead of division a/b, we compute the reciprocal of b and then multiply ab^{-1}.
- We take the convention that a/b = 0 when b = 0.