




Adaptive Control Law Development for Failure Compensation Using Neural Networks on a NASA F-15 Aircraft.


John J. Burken

NASA Dryden Flight Research Center, Edwards, CA

SAE Aero Tech 2005
October 3-6, 2005
Dallas/Ft. Worth Texas




1



Presentation Outline:

- Brief explanation of Generation II Flight Program**
- Motivation for Neural Network Adaptive Systems**
- Past/ Current/ Future IFCS programs**
- Dynamic Inverse Controller with Explicit Model Following**
- Types of Neural Networks Investigated**
- Brief example**
- Conclusions**



2



F-15 Intelligent Flight Control Systems

Motivation / Problem Statement {The Big Picture}

- Land a damaged airplane or, return to a safe ejection site.

General Goals & Objectives

- Flight evaluation of neural net software.
- Increased survivability in the presence of failures or aircraft damage.
 - Increase your boundary of a flyable airplane.
 - Increase your chances to see another day.
 - Increase your chances to continue the mission.



3

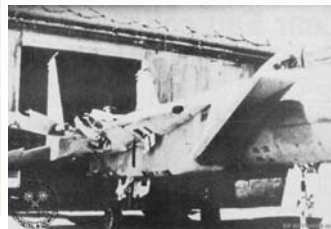


Motivation, cont

Airplanes in the Past Have Landed with Major Failures.

But not many!

Our Goal is to Increase the Survivability Region for the Pilot without luck or high skill levels or when the pilot is injured.



4



Past Flight Test of Reconfiguration Controllers

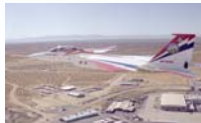
Flight Research Programs Not a Full List

- F-15 (Boeing,DFRC)
 - Flight Test in 1993
 - Simulated Failure : Stuck, Hardover, Missing Right Stabilator

- F-16 (Barron Associates, Inc.)
 - Flight Test in Mid 1996
 - Simulated Failure : Missing Left Horizontal Tail
 - Used real-time parameter identification

- X-36 (NASA Ames, DFRC, & Boeing)
 - Flight Test in December 1998
 - Jammed in-board elevon (15%)
 - Used Neural Networks to adapt to failure

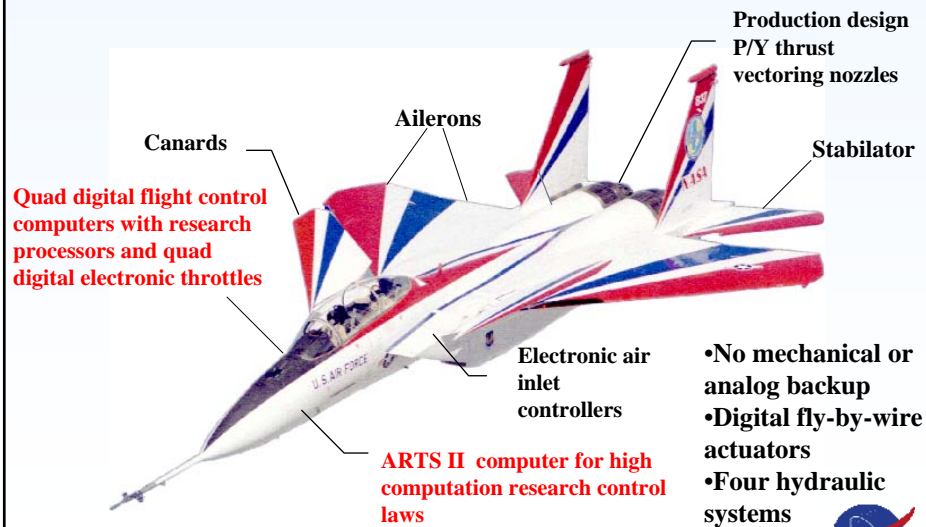
- F-15 (NASA Ames, DFRC, & Boeing)
 - Flight Test 1999 - present
 - Pre-Trained Neural Net (PTNN)
 - Used Neural Networks to organize real time stability derivative corrections into a database according to flight condition
 - Stabilator and Canard failure recovery using neural nets



5



NASA F-15 #837 Aircraft Description



6



General Neural Network Problem Statements

- Why Use a Neural Network?
- How much do Neural Networks help a controller?
- How much do Neural Networks cost w.r.t. compute power?
- How can we certify a Neural Network?



7

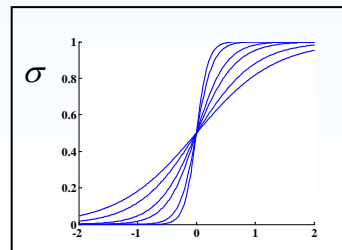
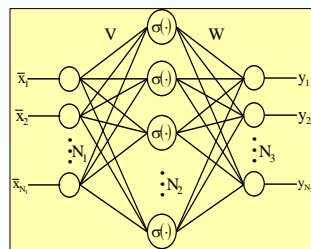


Why Neural Networks?

Neural Networks are Universal Approximators

Minimizes a H^2 norm

They permit a nonlinear parameterization of uncertainty



$$y = f(x) = W\sigma(Vx) + \varepsilon(x)$$

$$|\varepsilon(x)| < \varepsilon^* \quad \forall x \in \Omega$$

$$\dot{W} = -\left[\left(\sigma - \sigma' V^T \bar{x} \right) \eta + \kappa \|e\| W \right] \Gamma_W$$



8



Neurons in the human brain



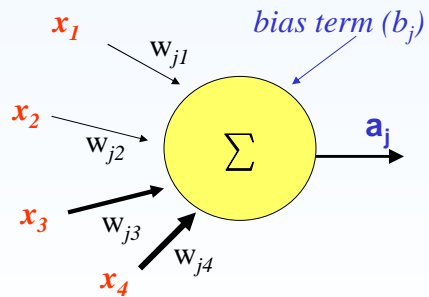
Neural networks simulate the activity of biological neurons within the human body. Neural networks are implemented in an attempt to re-create the learning processes of the brain by recognizing patterns.



9



Single Neuron



Combination function, $a_j = \Sigma w_{jk} * x_k + b_j$

summation product



10



General Adaptive Controller Statements

- **Two Types of Adaptive controllers**
 1. **Direct Adaptive**
 2. **Indirect Adaptive**
- **The Direct Adaptive Controller Works on the Errors.**
 - Needs a Reference Model to Generate $P_{err} = (P_{cmd} - P_{sensor})$
 - The Neural Network “Directly” Adapts to P_{err} .
 - Does not need to know the source of error.
 - No Aero Parameter Estimation Needed
- **The Indirect Adaptive Works on Identifying the source of Error.**
 - Does Not Need a Reference Model.
 - Needs to Identify the Aerodynamics that have changed! (PID)
 - PID is Time Consuming and *may not* be correct.



11

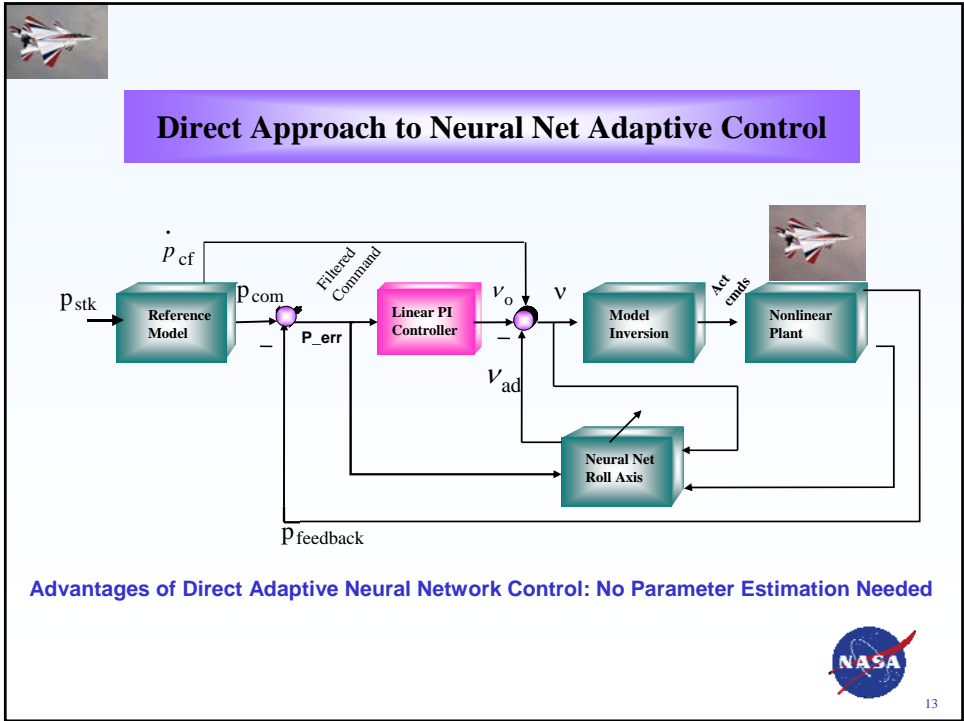



Background On Controller Types & NN

- NN have been intensively investigated with a dynamic inversion controller.
 - Many simulation test (F-15 / C-17 / Ames advanced cab ~B-757 ...)
 - One flight test with an unpiloted vehicle. (very limited X36 flight tests)
 - Very mature algorithms.
 - Relatively lower risk involved, compared to non-DI NN controllers.
 - Guarantee Bounds using the Lyapunov Function
- NN associated with non-DI controllers are just beginning to be investigated.
 - Not mature algorithms yet w.r.t. Neural Net side.
 - Note : We will refer to non-DI controllers as linear controllers.
 - Linear controls = (LQR, Root Locus, etc...)
 - Advantage of non-DI controllers: Legacy controllers on fleet.



12

Dynamic Inversion: 1st Step for our Adaptive Program

Consider the following non-linear system:

$$\ddot{x} = f(x, \dot{x}, u) \quad x, u \in \mathbb{R}^n$$

Recast as


$$\ddot{x} = v$$

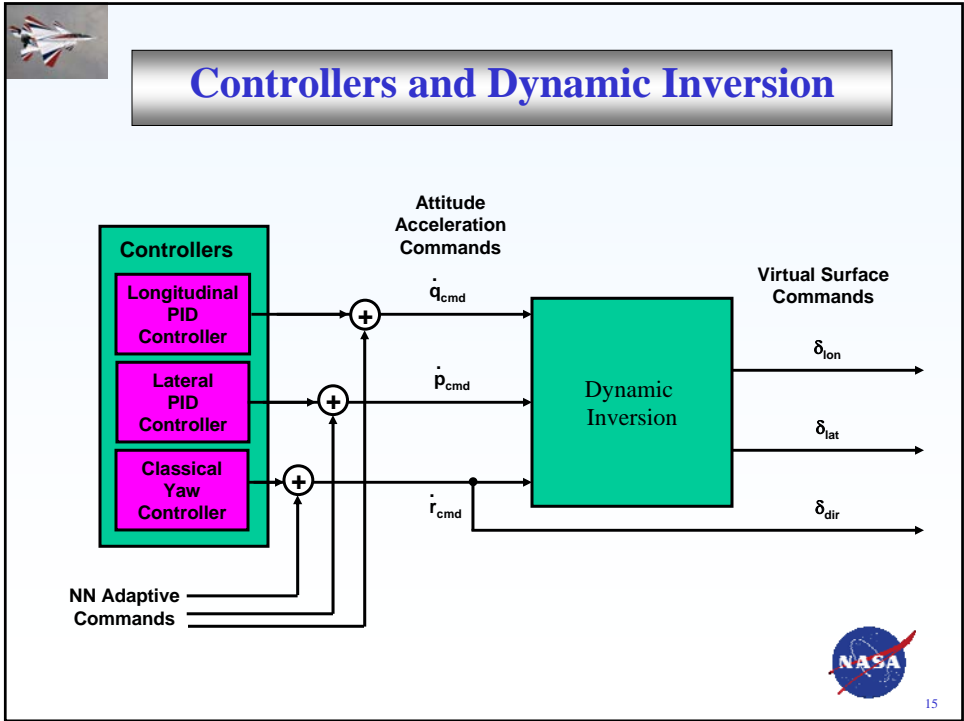
$$v = f(x, \dot{x}, u)$$

In the Ideal case,
Invert to obtain:

$$u = f^{-1}(x, \dot{x}, v)$$

Result: a linear time –invariant system





Classical Yaw Control

$\dot{\beta}$

Problem Statement & Background

- The Dynamic Inversion (DI) Method for Dir axis was not “well behaved with failures” With or Without the Neural Networks Active.
- Control Law Designers & Pilots were not happy with excessive beta excursions during pitch inputs with a stab failure.
 - This problem (... wiggles) was Not found for a Healthy / nominal airplane.

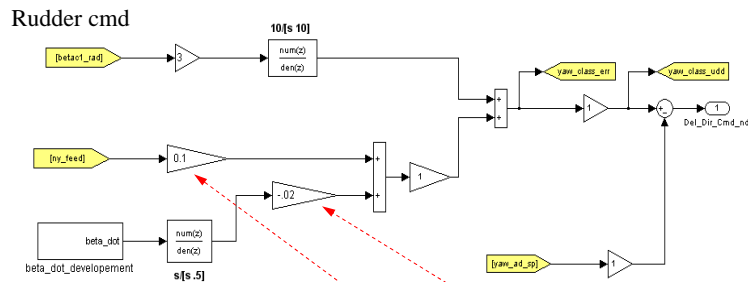
Solution Path Taken

- Tried Many DI modifications attempts.
 - I asked, “How about the classics” and use a hybrid system.
 - Use DI for the Pitch & Roll axes and Beta-dot for the Yaw axis.
- Decided on a simple beta-dot and Ny classical controller.

NASA
16



Classical Yaw Controller



Classical Yaw Controller Gains K_{ny} & $K_{\beta \dot{}}$ are changed to get desired dutch roll frequency and damping.



17



Calculation of Weights for Sigma-pi NN

$$\Delta W = -G(U_e B + L|U_e|W)\Delta t$$

ΔW are the weight calculations for the current time-step

W are the weights from the previous time-step

Δt is the time-step (0.0125 = 80 Hz rate)

G is the adaptation gain (or learning rate)

U_e is the from error compensation

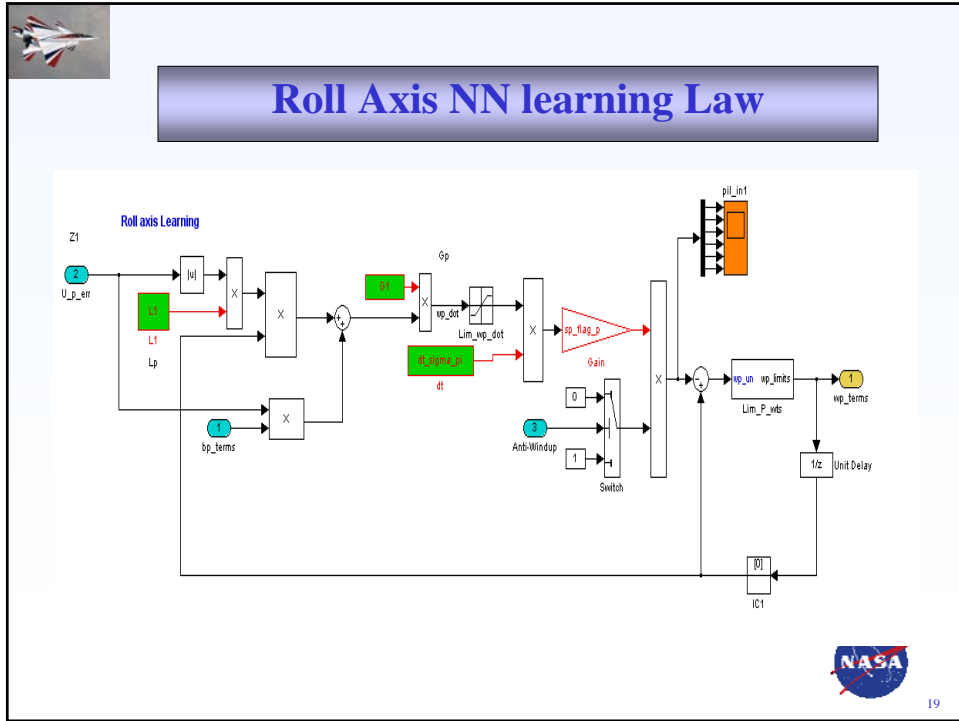
B are the basis functions

L is the deadband error to stop learning when error is small.

- Values for G & L are chosen (configurable constants)



18



Failures Investigated

2 groups of failures are “common” among aircraft mishaps/crashes.

- Aerodynamic Failures (A Matrix problems / lost aero surfaces, bent wings)
 - Canard Multiplier (changes lift).
- Control Failures (B Matrix problems / jammed control surfaces)
 - Right stab jammed at 6.85. deg





Neural Networks Investigated:

- Sigma-Pi (NASA Ames & Georgia Tech).
 - Chosen: Due to good cross coupling reduction.
- SHL (Single Hidden Layer, Georgia Tech).
 - Not Chosen due to lack of cross coupling reduction & time issues.
- RBF (Radial Basis Function, WVU).
 - Not chosen due to time issues.
- ADALINE (adaptive linear neuron network)
 - Not chosen due to time issues.

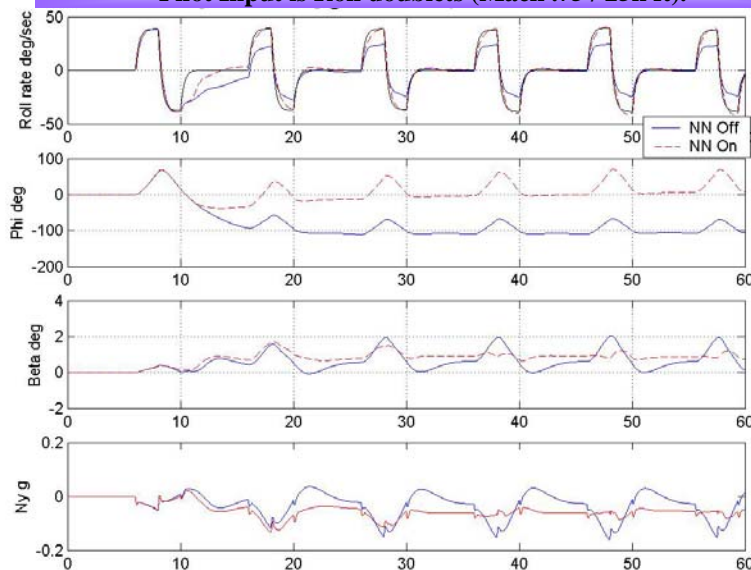


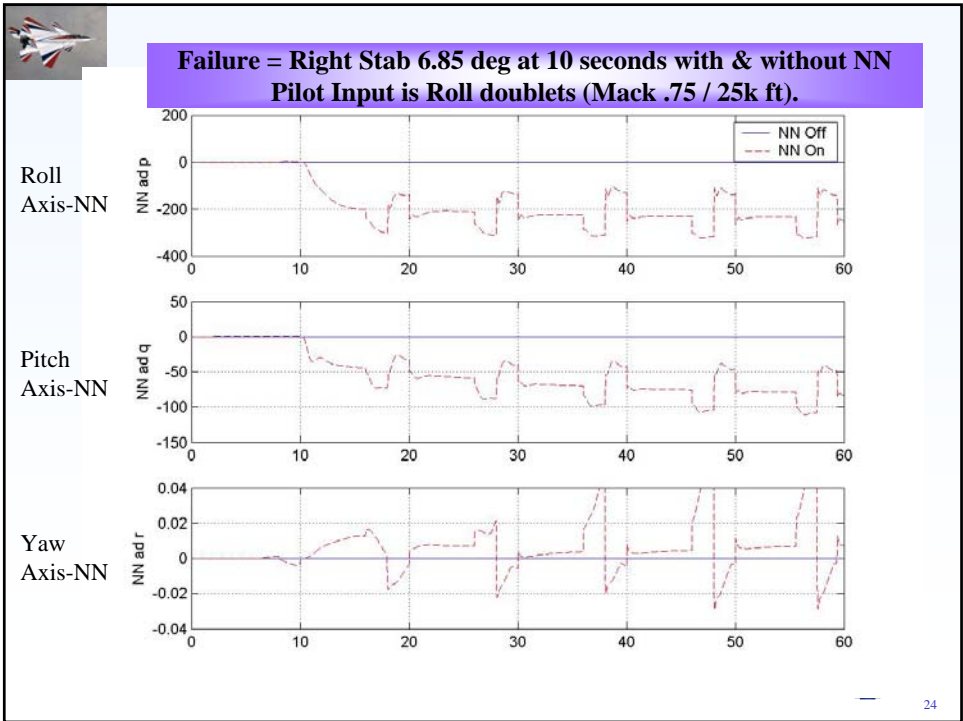
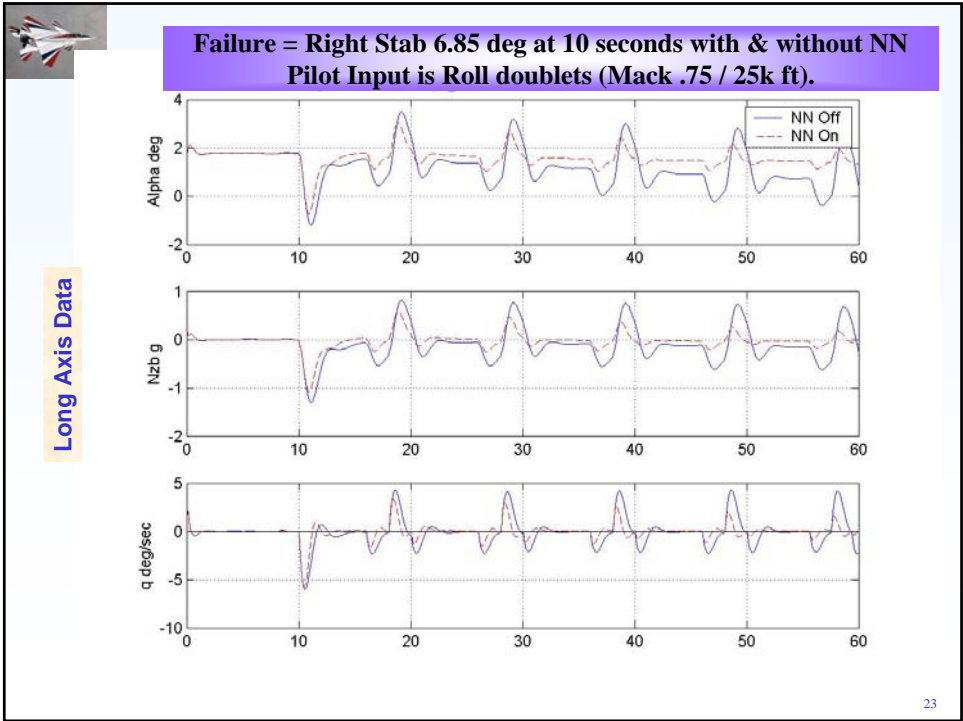
21

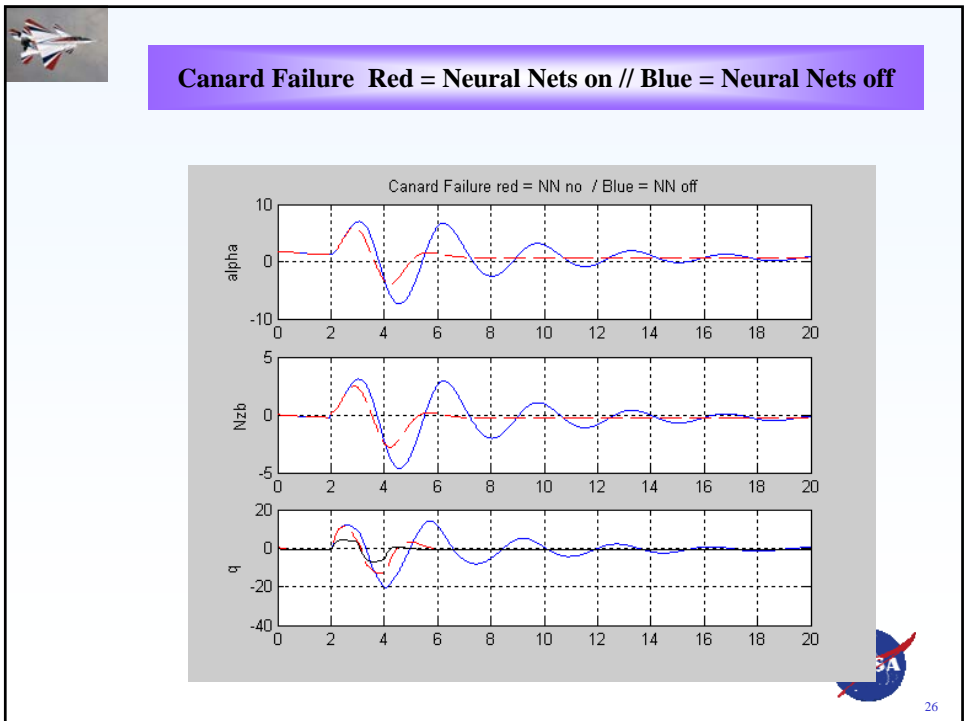
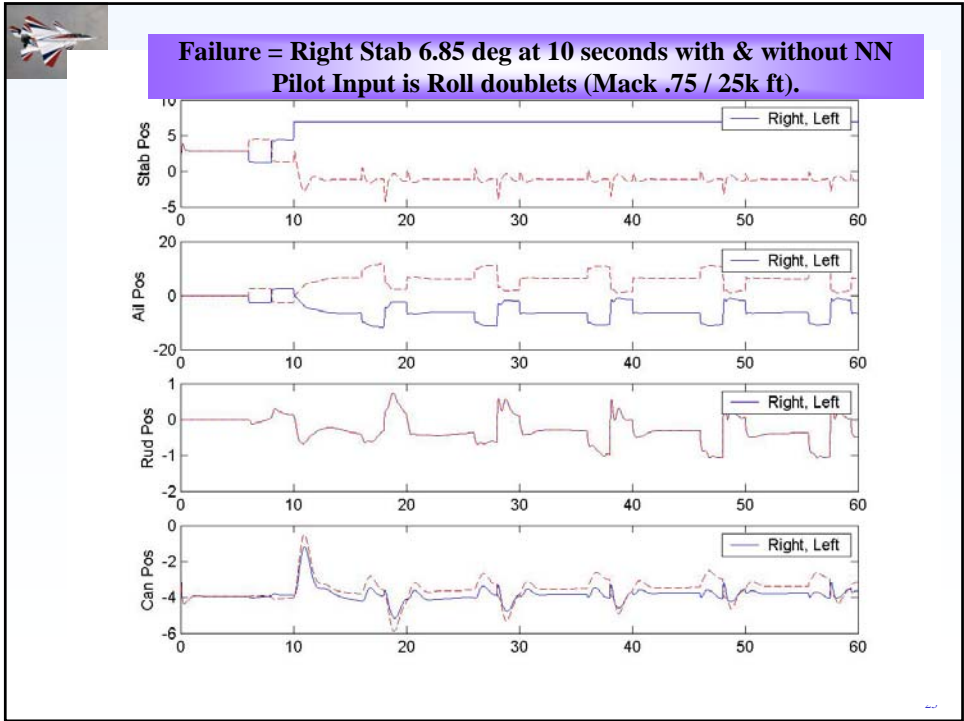


**Failure = Right Stab 6.85 deg at 10 seconds with & without NN
Pilot Input is Roll doublets (Mack .75 / 25k ft).**

Lat/Dir Axis Data









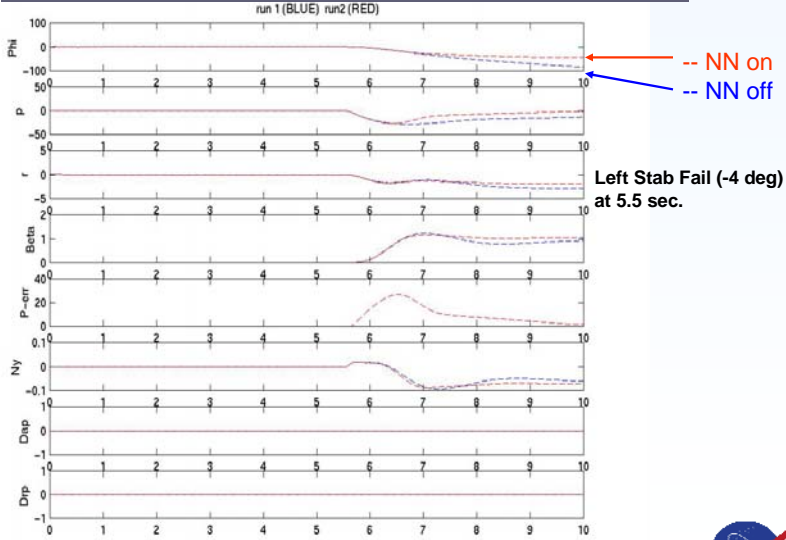
Problem Statement: NN Transients Analysis.

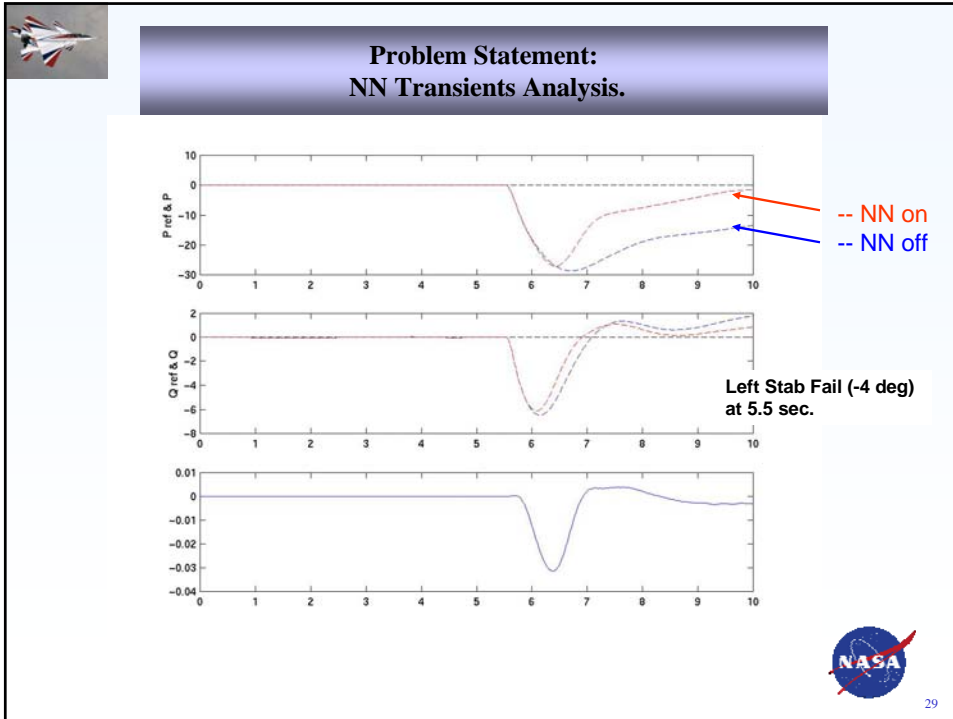
Problem Statement.

- With a failure in and Neural Nets On, are the transients smaller compared to the Neural Net off case.
- The following is a representative case
 - Left Stab Fail (-4 deg) at 5.5 seconds



Problem Statement: NN Transients Analysis.





-
- Summary / Comments**
1. Good tracking performance under failures.
 - Research Controller without NN is not to bad.
 - Research Controller with Neural Nets are better.
 - Failure Transients with Neural Nets on are smaller than without Neural Nets.
 2. Handling qualities are preserved (so-so)
 3. Reduction in pitch cross coupling due to roll inputs is accomplished.
 4. Hope to Flight Test Neural Network Controller October 05.
- NASA
- 30



Neural Network Cost

1. Some Neural Networks can be very computer intensive.
 - Sigma-Pi Neural Network is not time intensive.
2. How can we certify a Neural Network?
 - TBD



31



Backup Sides



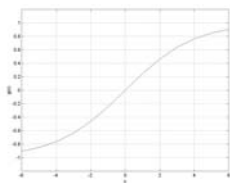
32



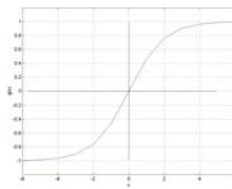
Squashing Function

Activation functions with a bounded range are called squashing functions

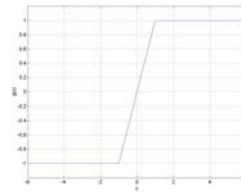
$$g(x) = \frac{1 - e^{-\text{gain} \cdot x}}{1 + e^{-\text{gain} \cdot x}}$$



gain = 0.5



gain = 1



gain = 10



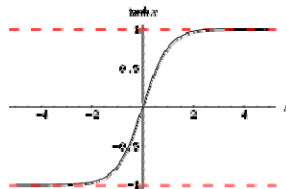
33



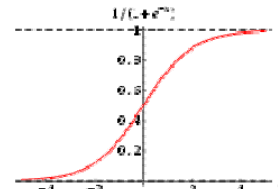
Activation Function

$g(a_j)$ is a non-linear function chosen by the neural network designer(s)

– Examples:



Hyperbolic tangent (tanh)



Sigmoid function



34

