2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Design of a Missile Autopilot using Adaptive Nonlinear Dynamic Inversion

Rick Hindman, Ph.D.
Raytheon Missile Systems
Tucson, AZ USA
Rick.Hindman@raytheon.com

William M. Shell
Raytheon Missile Systems
Tucson, AZ USA
wmshell@raytheon.com

***Abstract*— Traditional missile autopilot design typically uses a three loop feedback topology with gains dependent on the current flight condition of the missile. These gain values are obtained by interpolation on a predefined gain table. The gain values that make up this table are designed to balance performance and stability on the grid of flight conditions that define the missile's flight envelope. Robustness to parameter variation either requires a dense gain table, which necessitates a significant amount of on-board memory, or large stability margins, which may limit the aggressiveness of the missile's performance.**

This paper describes the continuing autopilot design project Adaptive Nonlinear Dynamic Inversion (ANDI). The ANDI autopilot uses non-linear dynamic inversion with an adaptive element to account for errors in the inversion process. A reference model is designed to provide the desired output performance. This technique allows the missile's performance to be tuned by simply adjusting a few reference model parameters. This results in a design that is robust with respect to aerodynamic modeling inaccuracies and to external disturbances.

I. INTRODUCTION

A missile is a highly agile system that exhibits nonlinear behavior. The typical missile autopilot is designed using system linearization and linear methods. Generally, no single autopilot design can stabilize the missile and provide the performance required over the entire desired flight envelope. To account for the changes in missile dynamics with varying flight conditions, a grid of design points is selected that covers the expected flight envelope. Unique autopilot gains are designed for the missile at each design point. This process can be both tedious and cumbersome. Typically, the gains for a given flight condition are designed using a linearization of the missile dynamics following a step command. Each autopilot design is then optimized to minimize the rise time and the command overshoot, while maintaining the required stability margins and flexible body and fin attentuation. These gains are then combined to create a gain table, which is examined point by point and spurious data is reexamined and adjusted. The autopilot gains used during flight are obtained by interpolating on this gain table for the appropriate gain values for the current flight condition. This method of autopilot design and operation is called gain scheduling. There are several problems encountered when using gain scheduled autopilots. The first is the assumption that flight condition changes slowly. When flight condition changes rapidly, such as for a missile with high thrust and/or high angle-of-attack flight, the resultant autopilot may not possess the stability properties of the linear control designs displayed statically at their local flight conditions. The second issue is that modifications are often made to a system during its service lifetime which require a redesign of the autopilot gain tables. Additionally, the payload of some systems may vary each time they are flown. Designing a gain scheduled autopilot for all contingencies would be extremely difficult.

The nonlinear autopilot design presented in this paper, ANDI, is based on the work presented in [1],[2]. ANDI does not involve gain scheduling. Instead it uses dynamic inversion (DI) to account for changes in the missile dynamics. Since DI is not robust to modeling errors [3], ANDI includes an adaptive control element in the form of an artificial neural network (ANN). The ANN is designed to correct those errors as well as other small magnitude errors due to wind, sensors, and/or modeling inaccuracies. A reference model is designed to provide an idealized closed-loop behavior for the missile system. The response of the reference model to the desired acceleration commands is measured, and the autopilot controls the missile to mimic that response. Pseudo control hedging (PCH) is added to avoid actuator saturation which may result in incorrect ANN learning [4]. PCH uses an actuator model with rate and position saturations to estimate the response of the fins to the fin commands. PCH maps the estimated fin deflections back to the estimated achievable control commands. The difference between the achievable and the commanded control levels can then be used to adjust the behavior of the reference model. This adjusted response is such that the autopilot will not require more of the actuators than they can provide. PCH adjusts the reference model when it identifies a response that is too aggressive for the actuators.

II. ANDI

The ANDI autopilot is designed in an attempt to control the missile over the entire flight envelope without gain scheduling. Typical autopilot designs balance stability against performance to provide a uniform design throughout the envelope. The step response at one flight condition will typically look very similar to the step response at another.

As flight conditions change it would be desirable to change the system performance appropriately. By varying parameters in the DI, the designer can control the shape of the response at different flight conditions. When the dynamic pressure is high, such as at low altitude and high Mach flight condition, missile response may be more aggressive than when the dynamic pressure is low. Gain scheduled design is motivated by the necessity to meet stability and performance requirements. Parameter scheduled design provides freedom to set system performance. Parameter selection, in general, will be a much simpler and less extensive task than designing a gain schedule. Ultimately, the desire is that either an analytic function, based on dynamic pressure, or a very sparse table can be used to provide the changing values of these parameters.

Figure 1 shows DI autopilot within a top level block diagram of the ANDI system.

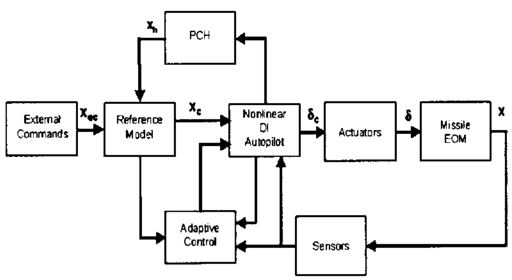

Fig. 1. ANDI Top Level Block Diagram

The "External Commands" block models commands from the guidance loop. These commands are issued to a reference model which then produces a desired command trajectory. This desired trajectory is the input to the "Nonlinear DI Autopilot", but this signal is also used by the ANN located inside the "Adaptive Control" block and by the "PCH" block. The ANN, trained using a Lyapunov learning rule, augments the DI autopilot in order to more accurately achieve the desired command trajectory. The PCH function adjusts the reference model, based on an estimate of the actuator capability, to avoid saturation. It prevents the reference model from commanding a trajectory unachievable by the system. This is necessary given the ANN's ability to change the signal produced by DI autopilot. The DI produces fin commands, δ_c, which are issued to the actuators to produce fin deflections within rate and deflection limits. The fin deflections pass into the equations of motion, "Missile EOM", to generate body angles, forces, and moments. The "Sensors" block observes the changes in the missile state and models the feedback information available to the autopilot.

The missile coordinate system used for this paper is shown in Figure 2. It complies with the industry standard aerodynamic body coordinate system.

This coordinate system is in motion relative to the inertial axis with the instantaneous velocity components U, V, and

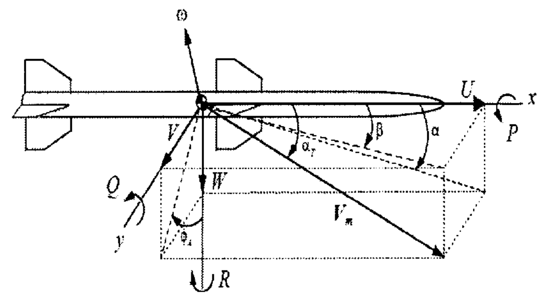

Fig. 2. Missile Coordinate System

W that align with the missile x, y, and z body axes. This missile coordinate system is also rotating relative to the inertial system with angular velocity components P, Q, and R around the x, y, and z body axes. Figure 3 is an expansion of the "Nonlinear DI Autopilot" block. The elements of this diagram will be discussed individually throughout this paper.

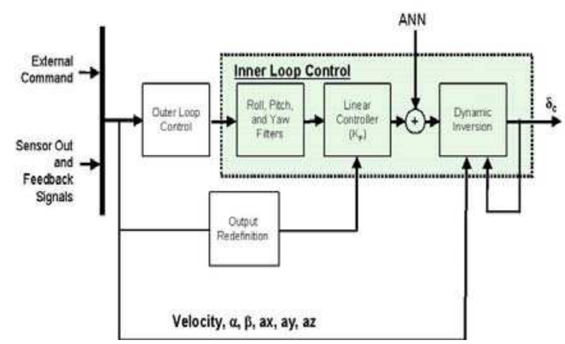

Fig. 3. Autopilot Block Diagram

A. Dynamic Inversion

Feedback linearization is one of the more popular forms of nonlinear control. It is a nonlinear coordinate transformation which recasts the nonlinear system into a linear time invariant (LTI) form allowing the designer to then apply classic linear control techniques. This may be performed on the entire system, or part of the system, using an exact state transformation and feedback. This method is unlike traditional linear techniques which use linear approximations of the plant dynamics. This technique presents the complication of requiring exact knowledge of the plant dynamics. Without such knowledge the system will have poor robustness [3]; however, this requirement can be overcome by including an adaptive element to account for inaccuracies. This will improve the robustness characteristics of the DI autopilot. It could also help reduce extensive (and expensive) aerodynamic modeling and wind tunnel testing.

DI is specific type of feedback linearization where the nonlinear plant dynamics are inverted and used as feedback [2]. Equation 1 is a nonlinear system description

$$\dot{x} = f + g \cdot u, \quad (1)$$

which is affine in u. If g is invertible, then the following control input would exactly cancel the system dynamics

$$u = g^{-1}[-f + \nu], \quad (2)$$

leaving

$$\dot{x} = \nu. \quad (3)$$

Typically dynamic inversion can only be applied to systems that are minimum phase [5]. Minimum phase indicates that a system has stable zeros, if it is linear, or that a system has stable zero-dynamics, if it is nonlinear. Non-minimum phase systems show an initial response in the "wrong" direction when the control is applied. Tail controlled missiles are non-minimum phase when acceleration at the center of gravity (CG) is used as a system output; therefore, they are not good candidates for DI control. By redefining the control output, one can create an output that is minimum phase allowing DI control to be used. This will be discussed further in Section II-C.2.

Figure 4 is an expansion of the "Dynamic Inversion" block within the "Inner Loop Control". The inputs to this block include the pseudo control signal, ν, given by

$$\nu = \nu_c + \nu_{ad}, \quad (4)$$

where

$$\nu_c = \dot{y}_c + k_{p\_inner}(y_c - y), \quad (5)$$

and ν_{ad}, the ANN output element, which will be covered in Section II-B.

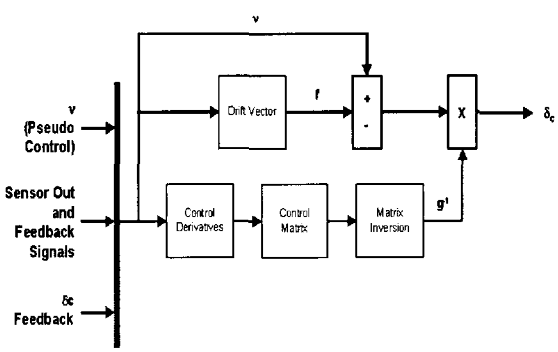

Fig. 4. Dynamic Inversion Block Diagram

Equations 6, 7, and 8 show the gain equations for the "Linear Controller" contained within the "Inner Loop Control". It is a proportional controller on the redefined output y, to be covered in Section II-C.2, with a feed-forward term, $\dot{y}_c$.

$$k_{p\_inner,P} = 3 \cdot \omega_P \quad (6)$$

$$k_{p\_inner,z} = 3 \cdot 2 \cdot \zeta_z \cdot \omega_z \quad (7)$$

TABLE I

GAIN EQUATION PARAMETERS

Gain Parameter	Value
ζ_z	0.8
ζ_y	0.8
ω_P	15
ω_z	10
ω_y	10

$$k_{p\_inner,y} = 3 \cdot 2 \cdot \zeta_y \cdot \omega_y \quad (8)$$

Table I are the ω and ζ parameters selected for this application. These parameters could be optimized and scheduled should the designer choose to do so. That possibility was not explored in this paper for two reasons. First, it was a desire to demonstrate an ability to reduce the scheduling and optimization requirements of the system. Second, the ANN should correct for suboptimal choices of these parameters.

B. Adaptive Neural Network

There are two methods of adaptive control: direct and indirect. The indirect method involves an adaptive system that produces estimates of system parameters. These evolving estimates are used by the control algorithms to continually update the controller parameters, which could be autopilot gains or something more advanced. Once the system successfully tracks the desired command, the adaptive autopilot is satisfied, even if the parameter values do not converge to their true values. If accurate estimates of the parameters are required, further effort will be required. With the direct adaptive method, estimates of system parameters are not made. Instead the algorithm adjusts the control parameters directly, which may not translate to physical system parameters at all. Again, the adapted parameters need not converge to any expected values, as long as proper command tracking is achieved.

One method of direct adaptive control uses Artificial Neural Networks (ANN). ANNs are widely used for their ability to accurately approximate continuous nonlinear functions. When used with DI, ANNs can help remove the effects of system and aerodynamic modeling inaccuracies [1]. This paper is restricted to multilayer feedforward networks. The weights of the various layers are trained using an online update law, designed using a Lyapunov stability proof [1]. The resultant system modifies the weights continuously and its output augments the computed control signal. This increases the robustness of the DI autopilot to uncertainties in the inversion process [2].

The "Adaptive Control" block, in Figure 1, consists of the Error Observer (EO) and the ANN as shown in Figure 5. The EO is discussed in Section II-C.4.

A common type of activation function used for the hidden layer neurons of such a network is the log-sig (9),

$$\sigma(\theta) = \frac{1}{1 + e^{-a\theta}}, \quad (9)$$

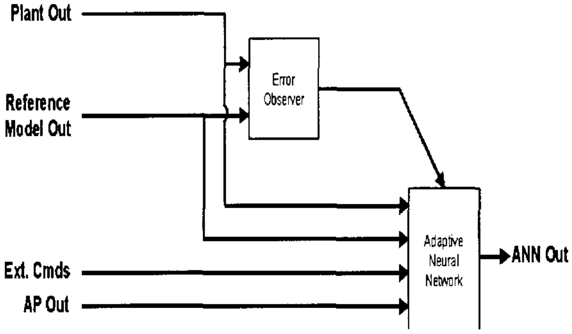

Fig. 5. Adaptive Control Block Diagram

where $a \in R$ is called the activation potential of the neuron. In this paper, multilayer networks with a single hidden layer of log-sig neurons and a linear output layer are used. For a network with n_1 inputs, n_2 hidden layer neurons, and n_3 output neurons, the value of the output layer of the neural network is given by the n_3 dimension vector,

$$\nu_{ad} = \varphi_{wk} + \sum_{j=1}^{n_2} w_{jk}\sigma_j(\theta_j), \tag{10}$$

where the activation of the hidden layer neurons is given by

$$\sigma_j(\theta_j) = \sigma\left(\varphi_{vj} + \sum_{i=1}^{n_1} v_{ij}x_i\right). \tag{11}$$

A top level block diagram view of the ANN is shown in Figure 6.

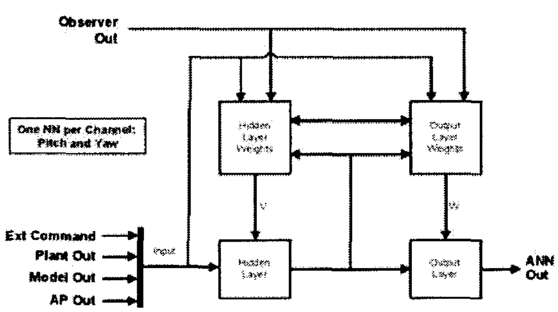

Fig. 6. Adaptive Neural Network

The values of φ_{wk} and φ_{vj} are weights multiplying constant bias terms (equal to one) affecting the activation level of the neurons. By defining the weight matrices and the activation vector as

$$\hat{V} \triangleq \begin{bmatrix} \varphi_{v1} & \cdots & \varphi_{vn_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix}, \tag{12}$$

$$\hat{W} \triangleq \begin{bmatrix} \varphi_{w1} & \cdots & \varphi_{wn_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix}, \tag{13}$$

TABLE II

ANN PARAMETERS

Gain Parameter	Value
k_e	0.1
Γ_v	1
Γ_w	1

$$\boldsymbol{\hat{\sigma}} \triangleq \begin{bmatrix} 1 & \sigma(\theta_1) & \cdots & \sigma(\theta_{n_2}) \end{bmatrix}, \tag{14}$$

the neural network output can be rewritten as

$$\nu_{ad} = W^T\boldsymbol{\sigma}(V^T\mu). \tag{15}$$

The variable μ is the input to the ANN with the constant 1 appended as the first term for the biases. The learning rule for $\hat{W}$ and $\hat{V}$ are given by the Lyapunov function derived equations [1, page 107].

$$\dot{\hat{W}} = -\Gamma_w[(\hat{\sigma} - \hat{\sigma}'\hat{V}^T\mu)\hat{E}^T P\bar{B}_m sgn(H_{\bar{u}}) + k_e||\hat{E}||\hat{W}] \tag{16}$$

$$\dot{\hat{V}} = -\Gamma_v[\mu\hat{E}^T P\bar{B}_m sgn(H_{\bar{u}})\hat{W}^T\hat{\sigma}' + k_e||\hat{E}||\hat{V}] \tag{17}$$

The variables Γ_v and Γ_w control the learning rates of the hidden and output layers. The variable k_e is used in the e-modification to ensure the ANN finds local solutions [1]. Table II contains the values of these variables used in this paper.

C. Supporting Technologies

1) Model Following: Ideally, for a given reference signal, the autopilot produces the needed fin commands for the missile to accurately follow that signal. Therefore, the reference model must be designed to meet system performance requirements within autopilot capabilities. The reference model provides the ideal closed loop behavior of the system. The reference models used in the ANDI autopilot are second order, in observability canonical form, one for the pitch channel and a second for the yaw channel. This form for the reference model is as specified in [1]. This is more restrictive than necessary. As long as the error observer is of the required form (second order, observability canonical form) then the ANN training laws will work as intended.

For a traditional system attempting to follow a step input, large errors are immediately observed by the autopilot. These errors will generate, through feedback signals, large control commands which drive the system to the commanded levels rapidly. When a continuous reference model replaces the step command, the initial errors observed by the autopilot will be small and will grow slowly producing a slow response. Therefore, in order to accurately follow a given reference model trajectory, the autopilot must provide lead using feed-forward signals of the given trajectory. This adds risk to the stability of the autopilot, but is required to provide the necessary impetus to initiate motion. Determining the appropriate feed-forward signal is an ongoing task.

2) Output Redefinition: Tail controlled missiles often use pitch acceleration, a_z, and yaw acceleration, a_y, as control variables. Because these variables are non-minimum phase, the autopilot must be separated into two elements, an inner loop and an outer loop [5]. The inner loop is minimum phase and suitable for DI while the outer loop maintains the non-minimum phase characteristics of the system. To bridge the two elements, the output of the outer loop is redefined in terms of appropriate minimum phase variables, such as α and q in the pitch channel or β and r in the yaw channel. Limiting this discussion to the pitch channel, the outer loop output is redefined as

$$y = \alpha + c_q q, \tag{18}$$

which acts as the input to the inner loop control. The parameter c_q is tunable and approximately equal to one. The DI pseudo control variable is defined as

$$\nu_c = \dot{y}. \tag{19}$$

The relationship between ν_c and δ_c is assumed to be of the form

$$\nu_c = f + g\delta_c. \tag{20}$$

The variable f is known as the drift vector and the variable g is the control derivative matrix, both of which are computed using estimates of the plant dynamics. Assuming g is nonsingular, Equation 20 can be inverted to obtain the command,

$$\delta_c = g^{-1}(f - \nu_c). \tag{21}$$

3) Outer Loop Control: The outer loop control stabilizes system accelerations using classical PI control techniques. ANDI uses PI control on the error signals; however, PI control is not strictly required. Any control methodology which provides a stable response and meets performance requirements may be used. The gains for the PI control are calculated using the desired ζ and ω of the outer loop transfer function. The outer loop control also includes a feed forward term based on the commands from the reference model. Additionally, the outer loop maintains the system characteristics, i.e., a non-minimum phase system is still non-minimum phase at this level.

The gains of the outer loop controller are given by

$$k_{p\_outer,z} = \frac{-(C_q \cdot \omega_z)}{2 \cdot \zeta_z \cdot Z_\alpha} \tag{22}$$

$$k_{i\_outer,z} = \frac{1}{C_q \cdot k_{p\_outer,z}} \tag{23}$$

$$k_{p\_outer,y} = \frac{(C_r \cdot \omega_y)}{2 \cdot \zeta_y \cdot Y_\beta} \tag{24}$$

$$k_{i\_outer,y} = \frac{-1}{C_r \cdot k_{p\_outer,y}} \tag{25}$$

where Z_α and Y_β are the partial derivatives of the accelerations with respect to body angle in the pitch and yaw channels.

4) Error Observer: The weight training laws of each artificial neural network require knowledge of the error between the model state and the missile state. However, only the missile output (acceleration) is available, not the state. Using an error observer driven by the output error, an estimate of the state error is obtained [1]. The error observer model is a LTI system based on the reference model dynamics. The output feedback gain matrix is designed such that the poles of the closed loop EO are two orders of magnitude greater than the reference model poles. Figure 7 shows the block diagram of the EO.

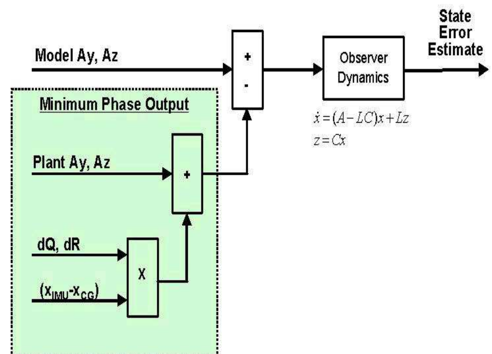

Fig. 7. Error Observer Block Diagram

In order to match the minimum phase characteristics of the reference model outputs, the plant outputs are converted to minimum phase. This is accomplished by moving the location of the measured acceleration forward of the missile CG.

5) Pseudo Control Hedging: A commanded control level, δ_c, may not be fully achievable due to fin rate or position saturation. The achieved actuator position, δ, if not measurable, can be estimated by modeling the response of the actuators to δ_c. The estimated control, $\hat{\delta}$, is used to calculate an estimate of the achieved pseudo control signal, $\hat{\nu}$, as

$$\hat{\nu} = f + g\hat{\delta}. \tag{26}$$

The pseudo control hedging signal, ν_h, is given by the difference between the commanded pseudo control, ν_c, and the estimated achieved pseudo control, $\hat{\nu}$.

$$\nu_h = \nu_c - \hat{\nu} \tag{27}$$

This is the amount of pseudo control lacking due to actuator saturations [4]. When the actuators are not saturated, the pseudo control hedge signal will be zero. Traditionally this signal would be used to "hedge" the model that outputs the command y. However, in the ANDI model, the command y is produced not by a reference model, but by the outer loop acting on the commanded and measured accelerations. In order to connect the hedged pseudo control signal to the acceleration reference model, a transformation using Z_α is used. Figure 8 is the block diagram of the PCH subsystem in its current configuration.

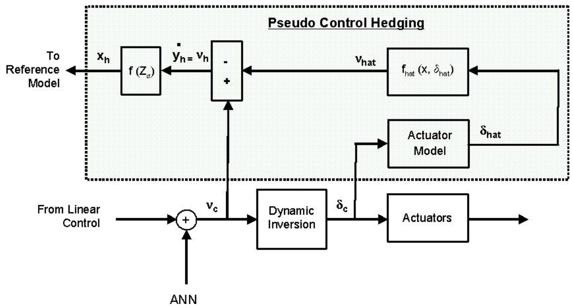

Fig. 8. Psuedo Control Hedging Block Diagram

III. RESULTS

Figure 9 shows the response of a classically designed gain scheduled autopilot and the ANDI autopilot in a fully coupled nonlinear missile simulation. A 7 G acceleration step command was initiated at 0.1 seconds in both pitch and yaw channels (yaw not shown). The response in the yaw channel exhibited similar characteristics. The dynamics of the reference model are given by the transfer function

$$RM(s) = \frac{136.13}{s^2 + 16.5s + 136.13} \tag{28}$$

The flight condition for this example is fixed at an altitude of 20 Kft and Mach 2.0, which represents a common condition for many air-to-air missiles. The reference model response to the 7 G step command is denoted by "RM" in Figure 9. The reference model natural frequency, ω_n, is 11.667 rad/sec and the damping ratio, ζ, is 0.707.

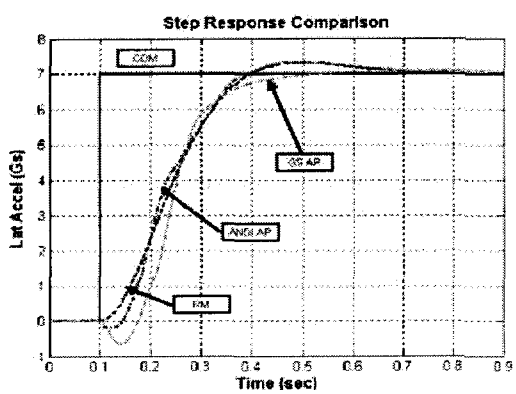

Fig. 9. Step Response of a Classic and ANDI Autopilots

Both autopilot responses show an initial "wrong" way response characteristic of non-minimum phase systems. The gain scheduled autopilot has a negative departure of approximately -0.65 Gs while the ANDI autopilot show a negative departure of approximately -0.2 Gs. The time constants of the two responses are nearly identical at 0.16 seconds. There are other quantitative measurements of performance, some that will favor the classic autopilot in this example, that fail to accurately express the true performance of these two systems. These quantitative measurements of performance will change for the ANDI system based on the chosen reference model while the gain schedules autopilot's response is fixed. The strength of the ANDI system is the flexibility of choosing the performance based on the needs of the system at the specified flight condition.

IV. CONCLUSION

This paper has presented the implementation of the ANDI autopilot on a skid-to-turn missile model. The commanded accelerations are processed by a reference model to provide the desired missile behavior to both the DI and the ANN. The ANN uses the response from the reference model and the DI autopilot to remove inversion errors and adjust the control signal to achieve the desired response. The autopilot is separated into an inner loop control and an outer loop control. The outer loop control maintains the non-minimum phase characteristics of the system while the inner loop control generates the minimum phase control signal used by the DI. PCH is used to adjust the reference model response when it would drive the system beyond the physical capabilities of the actuators.

To explore the benefits of this design, it was compared to a classic linearly designed autopilot. The results of this comparison are encouraging. Using the minimum phase output, the ANDI autopilot was able to track the desired reference trajectory accurately. The ANDI autopilot's non-minimum phase output follows the controlled minimum phase response closely. The primary difference in the responses is due to the initial 'wrong' way behavior characteristic of non-minimum phase systems. By adjusting the reference model, performance of the ANDI autopilot can be improved. Perhaps the most valuable quality is ANDI's robustness to aerodynamic modeling errors, which may reduce expensive and time consuming wind tunnel testing.

REFERENCES

[1] N. Kim, "Improved methods in neural network based adaptive output feedback control, with applications to flight control, Ph.D. dissertation, Georgia Institute of Technology, 2003.

[2] M. Sharma, "A neuro-adaptive autopilot design for guided munitions, Ph.D. dissertation, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, 2001.

[3] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*. New York, New York, USA: Springer-Verlage, 1999.

[4] E. N. Johnson and A. J. Calise, "Pseudo-control hedging: A new method for adaptive control, in *Workshop on Advances in Guidance and Control Technology*, Redstone Arsenal, Alabama, 2000.

[5] M. B. McFarland and S. M. Hoque, "Robustness of a nonlinear missile autopilot designed using dynamic inversion, 2000, aIAA-2000-3970.

[6] M. B. McFarland and A. J. Calise, "Robust adaptive control of nonlinear systems using neural networks, in *American Control Conference*, Albuquerque, NM, jun 1997.

[7] A. Isidori, *Nonlinear Control Systems*, 2nd ed. New York, Berlin, Heidelberg: Springer-Verlag, 1989.

[8] S. Haykin, *Neural Networks A Comprehensive Foundation*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice-Hall, Inc., 1999.

[9] H. Kaufman, I. Bar-Kana, and K. Sobel, *Direct Adaptive Control Algorithms Theory and Applications*. New York, Berlin, Heidelberg: Springer-Verlag Publishing Company Ltd., 1994.