

Real-time Stable Adaptive Control Implementation Using a Neural Network Processor

Timothy Robinson and Mohammad Bodruzzaman

Center For Neural Engineering, Tennessee State University
Nashville, Tennessee 37209-1561

Kevin Priddy and Karl Mathia

Accurate Automation Corporation
Chattanooga, Tennessee 37421

ABSTRACT

Helicopters are highly non-linear systems that have dynamics that change significantly with respect to environmental conditions. The system parameters also vary heavily with respect to velocity. These nonlinearities limit the use of traditional fixed controllers, since they can make the aircraft unstable. The purpose of this paper is to make contributions to the development of an "intelligent" control system that can be applied to complex problems such as this in real-time. Using a slowly changing model and a simplified nonlinear model as examples, a neural network based controller will be shown to have the ability to learn from these example plants and to generalize this knowledge for previously unseen plants. The adaptability comes from a neural network that will adjust coefficients of the controller in real-time while running on the Accurate Automation neural network processor.

Keywords: control systems, controllers, stability, neural networks, Youla parameterization, aircraft, real-time, parallel processing

1. INTRODUCTION

Stable aircraft control has been a desired goal for many design engineers for years. The idea of having in place a system that could respond to varying atmospheric conditions as well as to normal mechanical variations would give pilots an added sense of security while flying. This "intelligent" autopilot could serve as a backup and give the pilot greater control of his/her craft in tense situations.

This concept of a smart device is now possible using existing neural network paradigms and advances in hardware design. If the flight dynamics of an aircraft, for example a helicopter, are well understood, modeling data of its operation can be generated. This data can be processed through a neural network that can help generate an appropriate controller for maintaining stability. When this technique of stabilization is implemented on a high-speed neural network processing device, it can be applied to complex problems such as flight dynamics. This paper

will demonstrate the feasibility of this process in a real-time manner and draw conclusions about its effectiveness.

2. CONTROL THEORY

2.1. Conventional controllers

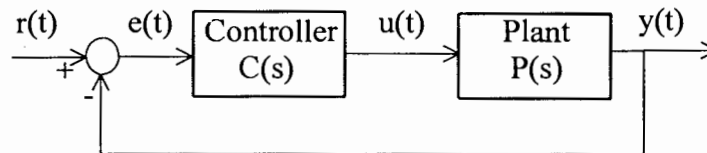


Figure 1. General control system.

A control system is defined as a system that exists for the purpose of regulating or controlling the flow of energy, information, or other quantities of a plant in some desired fashion. It is usually an interconnection of several components and in a closed-loop system information is exchanged between the input and the output.¹ Figure 1 contains the general symbolic description for the continuous case. For the discrete case, a similar diagram could be developed with the appropriate transformations from the S-domain to the Z-domain using the bilinear transform.

In this case, the aircraft dynamics have been sufficiently modeled by the plant equations and the appropriate controller has been designed to counteract any instabilities in this unchanging plant. The description of plant dynamics start with the input/output differential equation of the form

$$\frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_0 u(t) + b_1 \frac{du(t)}{dt} + \dots + b_m \frac{d^m u(t)}{dt^m} \quad (1)$$

Taking the Laplace transform, the above can be expressed as a polynomial transfer function in the s-domain as

$$P(s) = \frac{Y(s)}{U(s)} = \frac{b_0 + b_1 s + \dots + b_{m-1} s^{m-1} + b_m s^m}{a_0 + a_1 s + \dots + a_{n-1} s^{n-1} + s^n} \quad n \geq m \quad (2)$$

or an equivalent continuous state space representation in the controller canonical form of

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u \quad y = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} \quad (3)$$

where $x_1 = y$, $x_2 = \dot{y}$, ..., $x_n = y^{(n)}$ and $\dot{x}_1 = x_2$, $\dot{x}_2 = x_3$, ..., $\dot{x}_n = x_{n-1}$.

An appropriate controller can be found using either the method of pole placement or by applying the Linear Quadratic Regulator (LQR) design technique. The drawback with these controllers is that they are based on linearization of the plant and cannot take into account expected changes in mechanical operation or the disturbances exerted by environmental forces. A possible solution to this dilemma is to use Youla parameterization for controller design.

2.2. Youla parameterization

Youla (or Q) parameterization is based on the concept of fractional representation. The underlying idea is that every proper transfer function can be decomposed into the product of two stable transfer functions

$$P(s) = N(s)D^{-1}(s) = \tilde{D}^{-1}(s)\tilde{N}(s) \quad (4)$$

where N and D are right coprime and \tilde{N} and \tilde{D} are left coprime.² By finding the polynomials $U(s)$ and $V(s)$ which will satisfy Bezout's Equation

$$U(s)N(s) + V(s)D(s) = 1 \quad (5)$$

an expression for $Q(s)$ can be found which will satisfy the closed-loop expression

$$G(s) = U(s)N(s) - N(s)D(s)Q(s) \quad (6)$$

This $Q(s)$ can be used to construct the appropriate compensator from the following equation,

$$C(s) = \frac{U(s) - Q(s)D(s)}{V(s) + Q(s)N(s)} \quad (7)$$

Youla's theory guarantees that any controller developed by the factorization process in Equation 4 and also satisfies Equation 5 will be a stabilizing controller. The next step is to use the power of neural networks to generate the necessary coefficients of the $Q(s)$ polynomial.³

3. ARTIFICIAL NEURAL NETWORK

3.1. Feedforward algorithm

A multilayer feedforward neural network as shown in Figure 2 can be used to approximate many types of linear and nonlinear functions. When trained using error

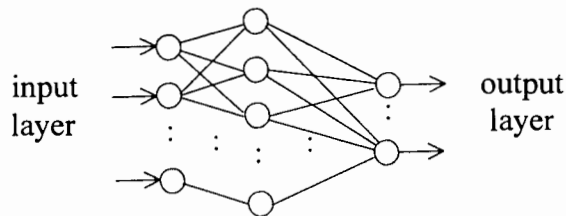


Figure 2. Multilayer feedforward network with nonlinear activation nodes and weights adjusted with error back-propagation.

back-propagation, input variables are mapped from one domain to another, thus allowing interpolation and estimation of the desired function. By exploiting this feature, coefficients for $Q(s)$ can be generated over a limited range.

3.2. Neural network processor

The Accurate Automation Corporation Neural Network Processor (NNP) is a multiple instruction, multiple data (MIMD) high speed parallel processor that has been optimized for neural network implementations. The NNP uses an instruction set that increases the efficiency of neural computations. By using an MIMD parallel processing architecture, one can update multiple neurons in parallel with an efficiency approaching 100% as the neural network size increases. This allows the NNP to operate in real-time with a speed over *140,000,000 connections per second*. (byte wide multiply/additions)

A simulation program was developed in C language that would download a previously trained neural network to the NNP and use it to generate the changed $Q(s)$ coefficients based on some changing parameter in the poles of plant $P(s)$.

4. SIMULATION RESULTS

4.1. Slowly changing model

The first simulation test used a plant with one changing parameter. The plant polynomial was $P(s) = \frac{(s+1)}{(s-\phi)(s+2)}$ where the value of ϕ took on values as shown in Figure 3. The neural network had been trained to compensate for ϕ values over the range of 2.5 to 3.5 with a nominal value of 3.0 and the plant response is to a step input. With the ϕ variable as input, the discrete coefficients of the polynomial $Q(z)$ are generated by the neural network. The graph shows that although the initial responses vary, the system is still stable.

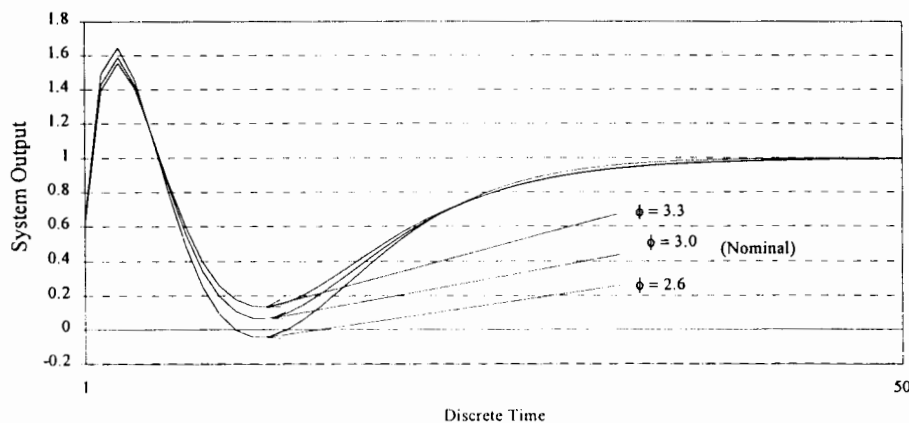


Figure 3. Simulation results for a slowly changing plant with one changing parameter, ϕ .

4.2. Simplified nonlinear model

The second simulation test used a plant with two changing parameters. The plant polynomial was $P(s) = \frac{(s+1)}{(s-\phi_1)(s+\phi_2)}$ where the values of ϕ_1 and ϕ_2 took on values as shown in Figure 4. The neural network had been trained to compensate for ϕ_1 values over the range of 2.5 to 3.5 with a nominal value of 3.0 and for ϕ_2 values over the range of 1.5 to 2.5 with a nominal value of 2.0. The plant response is to a step input. Again, with the ϕ_1 and ϕ_2 variables as input, the discrete coefficients of the polynomial $Q(z)$ are generated by the neural network. Note the disparity of the system outputs as either of the ϕ variables approaches the limits of the neural network data training set. The system outputs, however, still achieve a stable response.

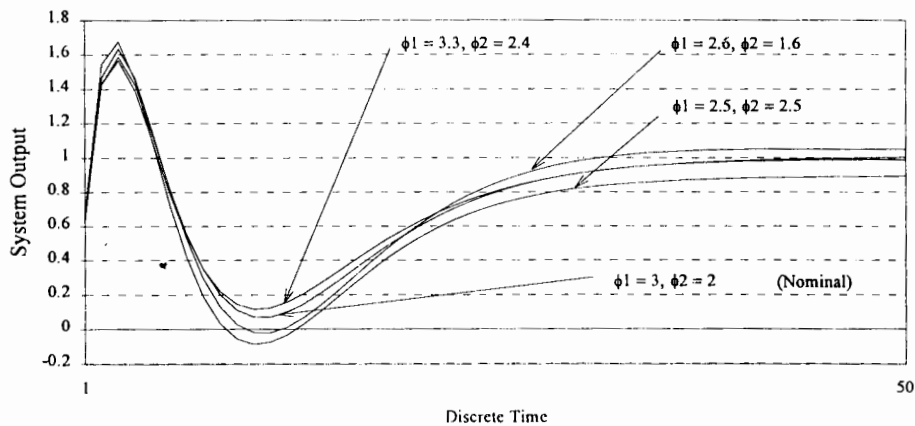


Figure 4. Simulation results for a slowly changing plant with two changing parameters, ϕ_1 and ϕ_2 .

5. CONCLUSIONS

The range of variability that can be tolerated by plant parameters must be predetermined in advance. This information is used by the neural network running on the NNP to determine the necessary Youla coefficients needed to produce the appropriate stabilizing controller. Since the neural network is trained off-line for parameter values over a specific range, values outside of this range are not guaranteed to produce the desired results. Therefore, it can be seen that this method works best for small changes over a narrow region.

This paper has demonstrated the viability of using the Youla parameterization technique for stabilization of simplified plants. The simulation program shows that a stable controller can be generated in real-time for slowly changing plant parameters. These slow changes are part of the behavior of aircraft such as helicopters. Although the plant description of a helicopter is much more complex, the stabilizing concept that has been presented remains the same.

6. ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research, which funds the Center for Neural Engineering at Tennessee State University. Additional support was supplied by Accurate Automation Corporation of Chattanooga, TN. The authors would like to thank them for their support in pursuing this research.

7. REFERENCES

1. Brogan, W., *Modern Control Theory, Third Edition*, Chapter 1, Prentice Hall, Englewood Cliffs, NJ, 1991.
2. Maciejowski, J. M., *Multivariable Feedback Design*, pp. 274-293, Addison-Wesley, Wokingham, England, 1990.
3. Desoer, C. A., Liu, R.-W., Murray, J. and Saeks, R., "Feedback System Design: The Fractional Representation Approach to Analysis and Synthesis," *IEEE Trans. on Automatic Control*, Vol. AC-25, No. 3, pp. 399-412, June 1980.