Adaptive Critic Flight Control

J.E. Steck[1] & R. Chandramohan[2]

[1]*Professor, Department of Aerospace Engineering, Wichita, Kansas 67260, USA*
[2]*Graduate student, Department of Aerospace Engineering, Wichita, Kansas 67260, USA*

1. Introduction

The study focuses on the design of an adaptive flight control system for a general aviation aircraft that is both robust and reconfigurable. The optimal design of a controller for a nonlinear dynamic system, such as an aircraft, required to minimize the cost is a challenging task as it necessitates the solution of a nonlinear two-point boundary value problem. Adaptive critics (AC's) provide a practical solution for the design of the controller to minimize the cost. AC's consist of two neural networks, an actor that approximates the optimal control law, and a critic that approximates the value or the cost-to-go function. The neural network parameters are varied such that they converge to the optimal solution over time. In this study classical and neural control systems are synthesized to combine the advantages of linear control theory and neural control. Initially the neural network architecture and parameters are determined from linear control theory. This part of the design is termed as the *pre-training* phase. In the next phase termed as the *online training* phase, the parameters of the neural networks are adapted to compensate for uncertainties not captured in the linearization as well as for control failures and parameter variations.

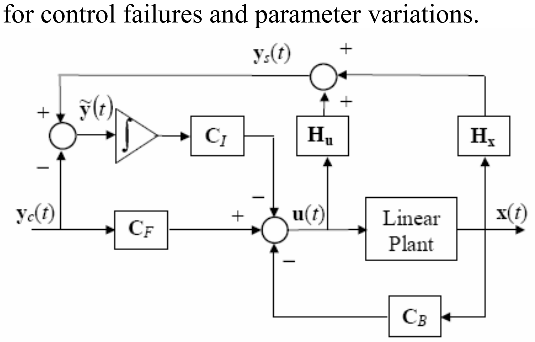

Figure 1: Linear PI controller

2. Design and Application

The Proportional-Integral (PI) controller is the multivariable linear control structure chosen to determine the neural network architecture and parameters for the *pre-training* phase. A PI controller, as shown in figure 1, modifies stability and provides type-1 response to command inputs [2]. PI controllers are designed for a set of equilibrium points called design points. The set of linear PI controllers provide performance targets to be matched by the equivalent neural network controller. The nonlinear model of the aircraft $\dot{x} = f[x(t), u(t)]$ is linearized about the set of equilibrium points to obtain LTI perturbation models about each equilibrium point, i.e. $\Delta\dot{x}(t) = F\Delta x(t) + G\Delta u(t)$. The objective of the linear PI controller is measured in terms of a quadratic cost given by

$$J = \lim_{t_f \to \infty} \frac{1}{2} \int_0^{t_f} [x_a^T(\tau) Q_a x_a(\tau) + 2x_a^T(\tau) M_a \tilde{u}(\tau) + \tilde{u}^T(\tau) R_a \tilde{u}(\tau)] d\tau$$

which is to be minimized with respect to $\tilde{u}$. $x_a \equiv [\tilde{x}^T \xi^T]^T$ is the augmented state. The weighing matrices in the quadratic cost are designed using *implicit model following* (IMF), to induce the closed loop system to follow the response of a model that satisfies established design criteria [2]. The forward gain C_F, the feedback gain C_B, and the Integral gain C_I which provide for optimal control in terms of the newly defined deviations ($\tilde{u}, x_a$) are obtained by solving the steady state Ricatti equation at each design point.

The PI Neural Network Controller (PINN) is the nonlinear structure motivated by the linear PI controller. It is obtained by replacing each linear gain with a nonlinear neural network, NN_B for C_B, NN_F for C_F, and NN_I for C_I, as shown in figure 2. The *Scheduling Variable Generator* (SVG) contains algebraic equation to generate the auxiliary inputs to

The neural networks. *The Command State Generator* (CSG) uses aircraft kinematic equations to generate the secondary elements of the state that are compatible with the commanded state.

The neural network control systems are synthesized by recognizing that the gradients of the nonlinear neural network correspond to the linear gains they replace at selected equilibrium conditions. The neural networks are feedforward neural networks with one hidden layer of sigmoidal functions. The number of neurons in the hidden layer is equal to the number of selected equilibrium points. The neural networks are initialized using a technique known as algebraic training of a neural network as given in [1]. This concludes the *pre-training* phase of the PINN.

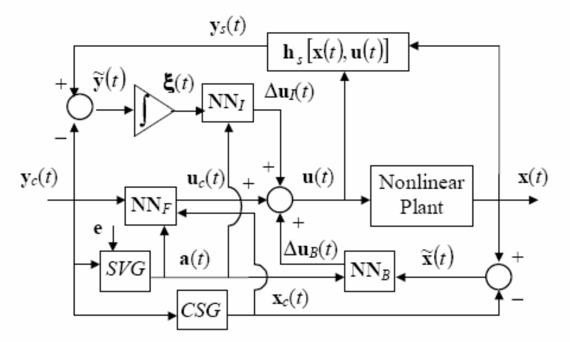

Figure 2: PINN controller

The PINN controller outlined above is designed for the longitudinal control of a general aviation aircraft. First the steady state flight envelope {*V, H*} is obtained based on stall speed, the power required and power available condition. A set of 36 equilibrium points are chosen inside and on the boundary of the steady state envelope. The aircraft is linearized about these operating points and an IMF linear PI controller is designed at these equilibrium conditions. Equivalent neural networks then replace the gains of the linear controller as outlined above. Each neural network contains 36 neurons in the hidden layer. The neural networks are then initialized using the algebraic training. The performance of the resulting PINN controller response is tested for a longitudinal command input ($y_c(t) = \{V, \gamma\}$) at a design point considered in the *pre-training* phase as well as at an interpolation point. The response of the neural network controller is compared to that of the linear controller specifically designed for the design point and the interpolation point.

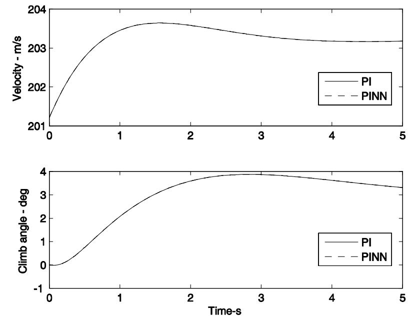

Figure 3: Response of PI and PINN controller for step input at a design point

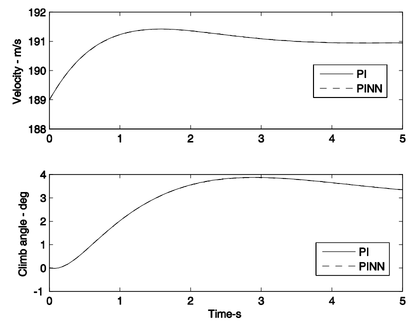

Figure 4: Response of PI and PINN controller for step input at an interpolation point

3. Conclusion

The response of the PINN controller, after the *pre-training* phase, at the design point matches exactly with that of a linear controller, furthermore the response of the PINN controller at an interpolation point is very close to that of the linear controller specifically designed for that point. The next step is to implement adaptive critic architecture to compensate for nonlinearities and control failures in the online training phase.

4. References

[1] S.Ferrari "Algebraic and Adaptive Learning in Neural control Systems," *A Dissertation*, (2002).

[2] C.Huang & R.F.Stengel, "Restructurable Control Using Proportional-Integral Model Following," *J. Guidance, Control, and Dynamics,* Vol.13, No.2, pp. 303-309, 1990.