



Model Reference Adaptive Flight Path Stabilization for Altitude and Velocity Hold

H. E. Taha^{*}, M. Bayuomi^{**}, G. M. El-Bayoumi^{***} and S. D. Hassan[†]

Abstract: The aim of this paper is to compare between different control techniques for the aircraft longitudinal motion. It deals specifically with poorly damped flight conditions. Three types of longitudinal autopilots are presented; rate of climb, altitude hold, and velocity hold autopilots. Some continuous-time control techniques are presented for the rate of climb autopilot; optimal control using state feedback, pole placement, and lag control. Finally, the *Model Reference Adaptive Control (MRAC)* technique is presented. The MRAC is selected to be the appropriate control technique for the autopilots of such types of aircrafts. Then It is used in designing an appropriate control system for the rate of climb autopilot, and also used in designing the control system for the rest autopilots; altitude hold and velocity hold. Due to the complexity of the velocity hold system for these types of aircrafts in some flight conditions, an online system parameter estimation algorithm is used along with an online design and implementation algorithms for MRAC.

Keywords: Altitude Hold Autopilot, Velocity-Mach Hold Autopilot, and Model Reference Adaptive Control

1. Introduction

This paper is to design stability and control augmentation systems for high speed fighter aircrafts, where the very poor dynamics exist. The FOXTROT a/c (a twin engined, jet fighter/bomber a/c) was selected to be the example a/c, in this paper, for designing the flight control system. The data of this a/c is available at four flight conditions, [1]. For every flight condition a number of autopilots are to be designed; a velocity autopilot and a rate of climb autopilot followed by an altitude hold one. In this paper, the rate of climb autopilot is designed using continuous-time control techniques; optimal control using state feedback, pole placement, lag control, and finally using a discrete-time control technique (MRAC). The designed control algorithms are then applied to the aircraft nonlinear equations of motion model. The MRAC technique showed the capability versus the other control techniques for fulfilling the desired response characteristics, so It is selected for designing the altitude hold and velocity hold autopilots. But in the velocity hold autopilot, the MRAC stand alone did not show the same capability for control as in the rate of climb and altitude hold autopilots; this is because the complexity of the velocity hold autopilot in some flight conditions. So an online system parameter estimation algorithm is used along with an online algorithm for MRAC design and implementation.

^{*} Teaching Assistant, Aerospace Eng. Dept., Cairo University / egyhezzat@gmail.com

^{**} Assistant Professor, Aerospace Eng. Dept., Cairo University / msb0100@yahoo.com

^{***} Professor, Aerospace Eng. Dept., Cairo University / gelbayoumi@yahoo.com

[†] Professor, Aerospace Eng. Dept., Cairo University / sayedhassan@hotmail.com

2. Rate of Climb (ROC) Autopilot

To change the altitude from a level to another, this should be done using a rate of climb command not an altitude command.

To design a linear controller for a nonlinear system, two criteria has to be specified; the desired response characteristics (settling time, overshoot...) and the maximum value of the command input that would be required from the system. The maximum ROC of the FOXTROT a/c is 50 m/sec. The criterion used along this paper is to achieve a critically damped nonlinear response with 20 sec settling time for the maximum value of the a/c ROC.

2.1. Rate of Climb Continuous-Time Autopilot

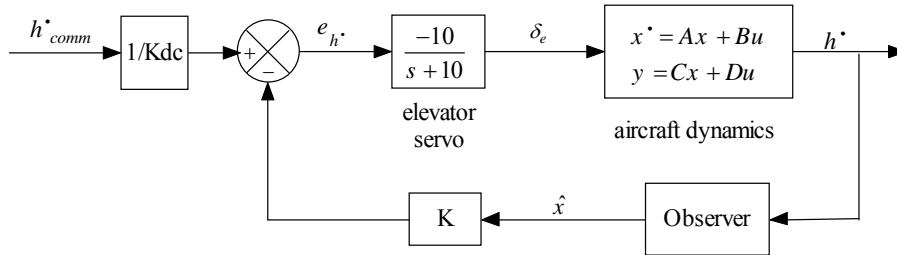


Fig. 1 Rate of Climb Autopilot using State Feed Back

Flight condition (3) of the FOXTROT a/c, 350 m/sec velocity and 10 km altitude [1], is used when designing the ROC autopilot. The state equation of the elevator servo, [2,3] is incorporated along with both the linear state space model and the nonlinear equations of the a/c longitudinal motion [4,5]. The open loop transfer function, (Fig. 1), is given by:

$$\frac{\Delta h^{\bullet}}{\Delta e_{h^{\bullet}}} = \frac{207.1 s^3 - 127.9 s^2 - 54030.0s - 630.7}{s^5 + 11.83 s^4 + 46.81 s^3 + 285.1 s^2 + 4.389 s + 0.7362}, \quad K_{dc} = -856.7 \frac{m/sec}{volt} \quad (1)$$

where 1 volt is corresponding to 1 rad elevator deflection.

The open loop response for a 50 m/sec ROC command is shown in Fig.2, which is the maximum ROC for the FOXTROT a/c at flight condition 3. It comes from $50/K_{dc} \Delta e_{h^{\bullet}}$ command (about 3.34° upward elevator).

Figure 2 shows that the open loop response is very poor. Also note the difference between the linear and nonlinear steady state values. This is because to get 50 m/sec ROC, a $50/K_{dc}$ input is given to the system. Since the dc gain is constant for the linear system, the linear result settles at 50 m/sec exactly. But the dc gain is not constant for the nonlinear system, so the nonlinear result did not reach the desired ROC value.

A state feedback control will be applied as shown in Fig.1 to enhance the system response. The gain matrix K can be designed either using any optimal control technique (e.g., Riccati) or using the pole placement technique (Ackermann) by suggesting appropriate locations for the closed loop poles.

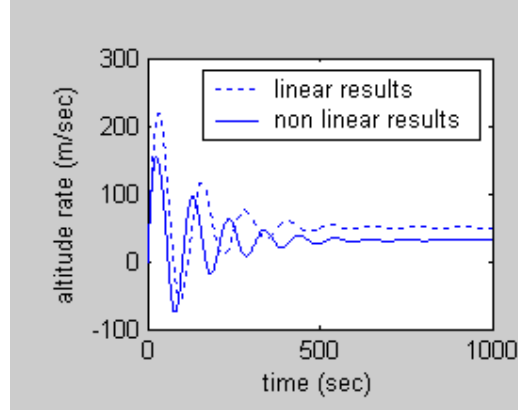


Fig. 2 ROC open loop response to 50 m/sec ROC command.

2.1.1. ROC Continuous-Time Autopilot Using Riccati Technique

This section applies Riccati technique which works to minimize a certain performance index function: $J = \int (x'Qx + u'Ru)dt$, [6]. To get the gain matrix K that minimizes the stated J, the following selection for Q and R is used:

$$Q = \begin{bmatrix} 1/u_{\max}^2 & 0 & 0 & 0 & 0 \\ 0 & 1/w_{\max}^2 & 0 & 0 & 0 \\ 0 & 0 & 1/q_{\max}^2 & 0 & 0 \\ 0 & 0 & 0 & 1/\theta_{\max}^2 & 0 \\ 0 & 0 & 0 & 0 & 1/\delta_{e_{\max}}^2 \end{bmatrix}, R = \begin{bmatrix} 1/e_{h_{\max}}^2 \end{bmatrix} \quad (2)$$

where

u_{\max} is determined from the engine thrust and a/c performance

q_{\max} and θ_{\max} are specified as desired according to the a/c operating limitations

$\delta_{e_{\max}}$ is determined from the elevator mechanical restraint, $\delta_{e_{\max}} = 25^\circ$, $w_{\max} = u_0 \tan \alpha_{\max}$;

Condition of stall. Choose $\alpha_{\max} = 25^\circ$

$e_{h_{\max}}$ is the maximum output limit of the device performing summation for $(h_{comm}^{\bullet} - k \Delta x)$

which is usually an op-amp, so it is chosen to be 15 volts.

w_{\max} , $\delta_{e_{\max}}$, and $e_{h_{\max}}$ are only considered in this analysis with the stated values. The closed loop response for a 50 m/sec ROC command is shown in Figs. 3 and 4.

It is clear that the state feedback using Riccati technique enhanced the response. But it is still very poor (200% POS, 350 sec Ts). On the other hand, the nonlinear response was unstable even for small values of the ROC command. This response can be enhanced more by tuning the Q and R matrices, but such tuning is blind, since the effect of Q and R on the closed loop poles and consequently the time response is not specifically known.

2.1.2. ROC Continuous-Time Autopilot Using Ackermann's Technique

This section uses the pole placement technique to locate the closed loop poles at appropriate locations; critically damped response is required with about 20 sec Ts, then the dominant closed loop poles are located at -0.2, and the remaining poles are suggested to be real at -0.6, -1.0, -1.4. The gain matrix K can be calculated using Ackermann's Formula, [6]. Figures 5 and 6 show the closed loop response for 5 m/sec ROC command, below the instability limit.

Placing the closed loop poles at the stated locations enhanced greatly the settling time, but due to this rapid response the POS has increased to about 500%.

The system nonlinear response deteriorates with increasing the ROC command and became unstable for 10 m/sec ROC command. The response for 9 m/sec ROC command (instability limit) is shown in Fig. 7.

Introducing the command input in a gradual form like $h'_{comm} = ROC_{comm}(1 - e^{-t/\tau})$ instead of the step form may increase the ROC limit above 9 m/sec. It is found that this does not increase the ROC limit but it delays the time at which the instability occurs. And the larger τ , the more delay for instability, as shown in Figs. 8 to 10. This delay if increased, the system can be considered stable.

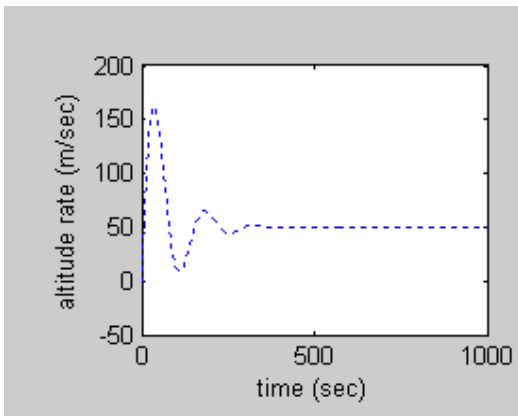


Fig. 3 ROC linear response to 50 m/sec ROC command, using Riccati.

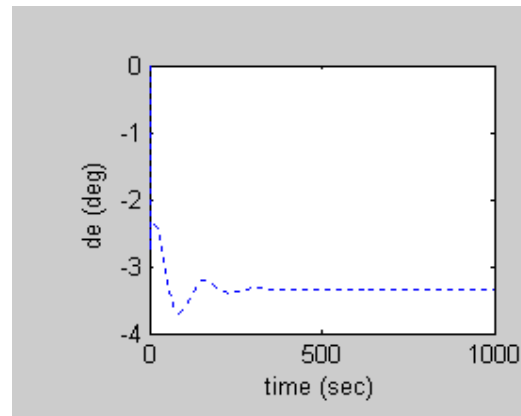


Fig. 4 Elevator deflection to 50 m/sec ROC command, using Riccati.

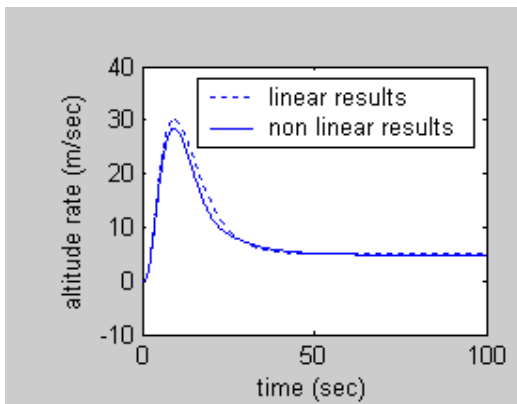


Fig. 5 ROC critically damped response to 5 m/sec ROC command, using Ackermann

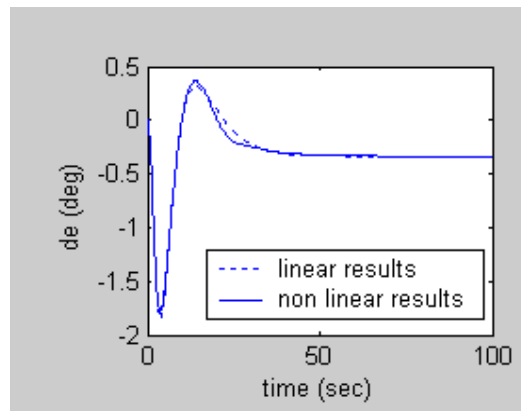


Fig. 6 Elevator deflection for critically damped response to 5 m/sec ROC command, using Ackermann.

2.1.3. ROC Continuous-Time Autopilot Using Classical Control

There are two problems; one concerning the very peaky response, and another concerning the instability occurring in the nonlinear response as the ROC command increases. They are related together. So if the first problem is solved, the second problem would vanish or the instability limit (9 m/sec) would increase. It is clear that the very high peak is not due to the poor damping of the dominant poles but from a poor zero at -0.0117; very close to the origin.

This poor response can be enhanced using lag control; Fig. 11, putting the lag controller pole so that to cancel the zero near the origin, and the lag controller zero so that to make the system critically damped at $S = -0.2$ to give the required $T_s = 20$ sec.

The result is:

$$G_c = 0.0168 \frac{1+10s}{1+84s}$$

The root locus was plotted for negative gain because the existence of the zero in the right half plane, Fig. 12.

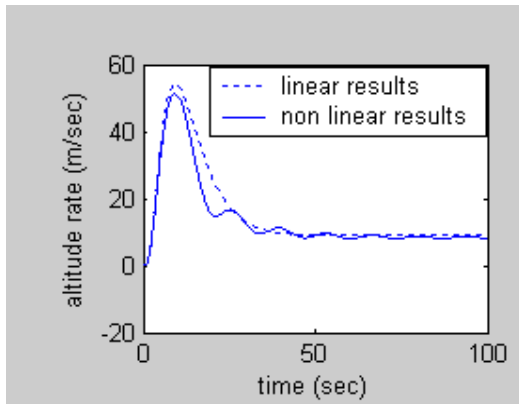


Fig. 7 ROC critically damped response to 9 m/sec ROC command, using Ackermann

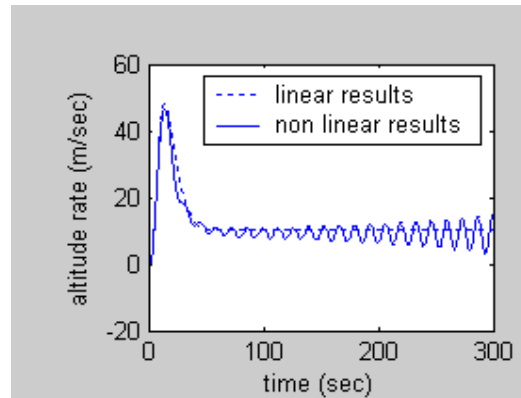


Fig. 8 ROC critically damped response to 10 m/sec ROC command, $\tau = 5$ sec

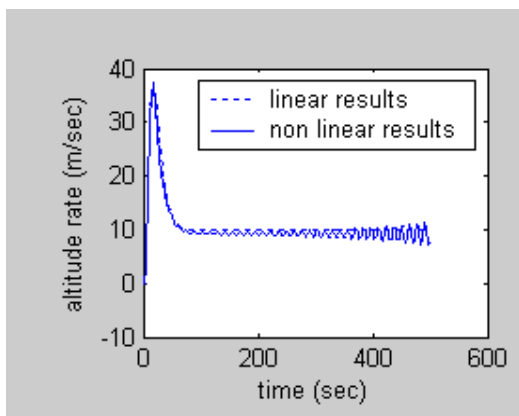


Fig. 9 ROC critically damped response to 10 m/sec ROC command, $\tau = 10$ sec

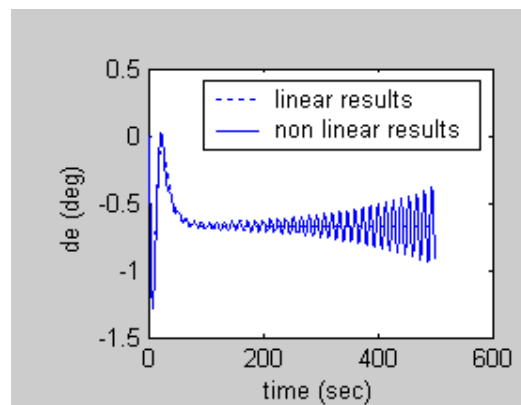


Fig. 10 Elevator deflection for critically damped response to 10 m/sec ROC command, $\tau = 10$ sec

Figures 13, 14, 15, and 16 show the closed loop response for 20 m/sec ROC command, below the new instability limit.

As expected, both overshoot and settling time have greatly enhanced, there is approximately no overshoot and the settling time for the controlled variable (ROC) is about 20 sec. This result leads a paradox; "how the lag controller, which is considered to be a poor controller (having only 3 tuning parameters), provides much better results than the state feedback control technique, which is considered to be one of the most powerful control techniques nowadays (having 5 tuning parameters in our case)!". This statement is valid only for this case due to the presence of a poor zero near the origin. This zero could be eliminated using the lag controller while the state feedback control technique cannot handle the system zeros.

The settling time for the ROC and altitude is different from that for the angle of attack, flight path angle, elevator deflection, and the a/c speed. This is because the poor zero near the origin exists only for the ROC and altitude transfer functions. This zero is cancelled by the lag controller pole, and the settling time was as designed. But for the other stated variables this poor zero doesn't exist, hence the pole from the lag controller was not cancelled, and also became the dominant one for these variables, leading to over damped response with very big settling time. But this big settling time will be handled when designing another simultaneous controller for these other variables (velocity autopilot).

Also for the second problem concerning the instability occurring due to high values of the command input, as expected, the ROC limit has increased much. The autopilot starts to reach instability for ROC command greater than 35 m/sec. The response to 35 m/sec ROC command is shown in Fig. 17. On the other hand, the response for 40 m/sec ROC command reaches instability after more than 600 sec which is considered to be stable as stated earlier.

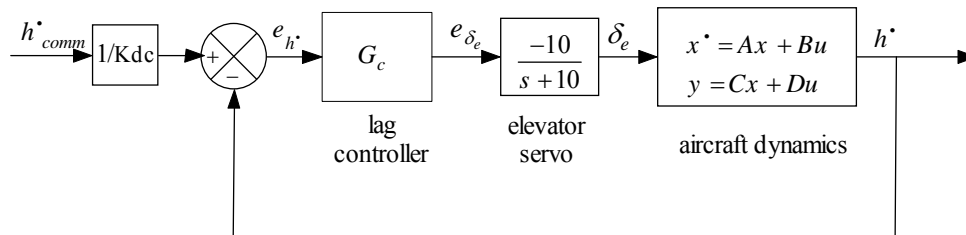


Fig. 11 Rate of Climb autopilot using classical control techniques (lag controller)

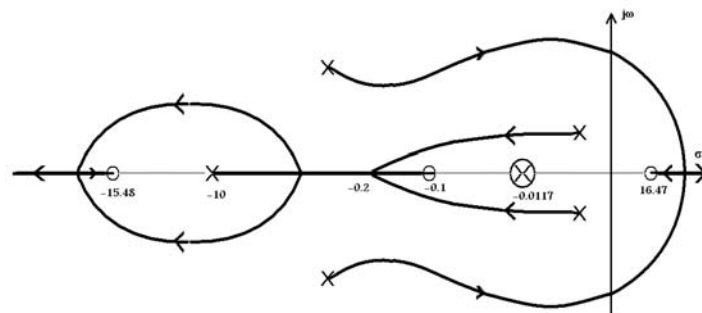


Fig. 12 Root locus plot for the a/c dynamics, elevator servo, and the lag controller

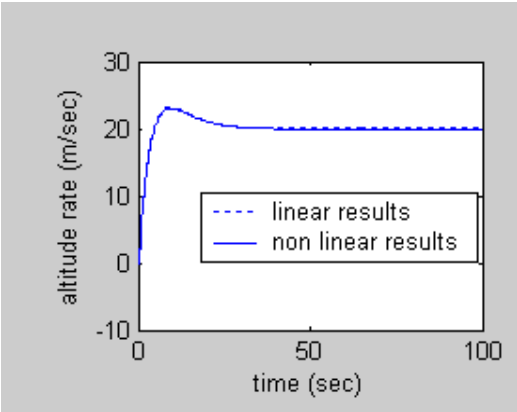


Fig. 13 ROC critically damped response to 20 m/sec ROC command, using lag controller.

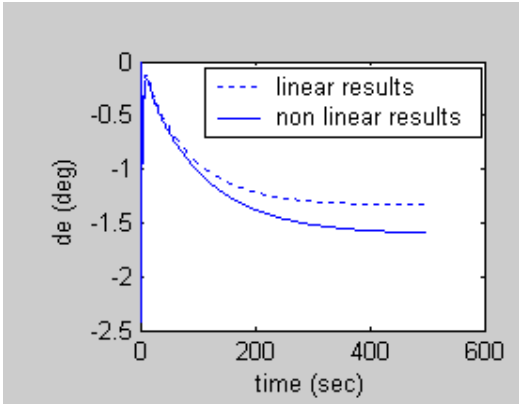


Fig. 14 Elevator deflections for critically damped response to 20 m/sec ROC command, using lag controller.

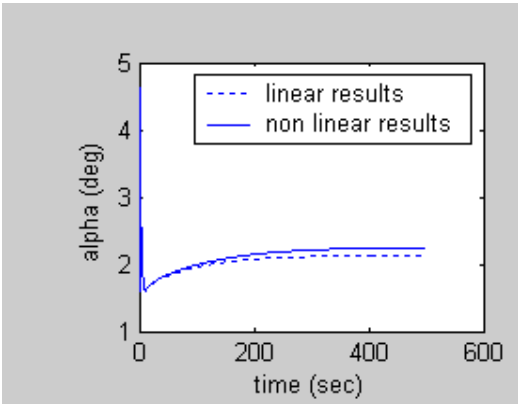


Fig. 15 Angle of attack critically damped response to 20 m/sec ROC command, using lag controller.

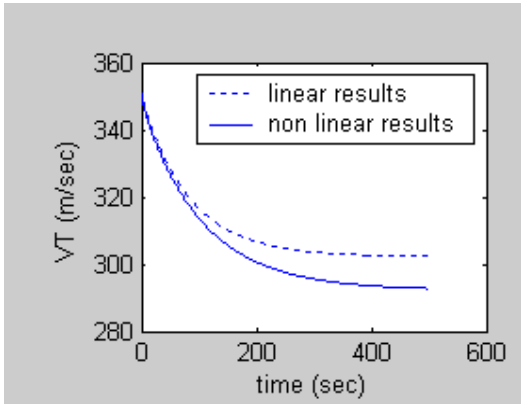


Fig. 16 Velocity critically damped response to 20 m/sec ROC command, using lag controller.

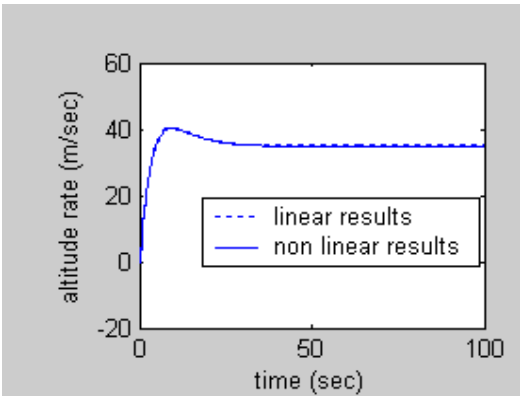


Fig. 17 ROC critically damped response to 35 m/sec ROC command, using lag controller

The response has been enhanced very much and reached appropriate values for the ROC. But the criterion specified before requires a critically damped nonlinear response with 20 sec Ts for the maximum ROC of the example a/c (50 m/sec). This is not satisfied in the designed system. And it can be achieved by using adaptive control techniques like the controller scheduling technique or by using MRAC. But the controller scheduling technique will result in a slower system, because of the addition of the two settling times for the two controllers.

So, the MRAC technique (digital autopilot) is to be used to attain this value for the maximum ROC (50 m/sec) through an appropriate nonlinear response and keeping stability for the a/c.

2.2. Rate of Climb (ROC) Discrete-Time Autopilot

The block diagram of an MRAC for controlling the a/c ROC is shown in Fig. 18. The discrete transfer function representing the a/c dynamics and elevator servo can be determined from the continuous one (equation 1), and using an appropriate sampling period, $T = 0.1$ sec.

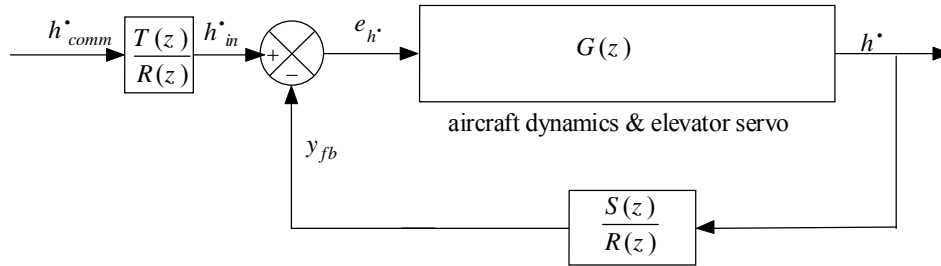


Fig. 18 Digital Rate of Climb autopilot using MRAC

$$\frac{h^{\cdot}(z)}{e_h^{\cdot}(z)} = \frac{-0.5014 z^4 + 3.005z^3 - 1.061z^2 - 1.832z + 0.3934}{z^5 - 3.946z^4 + 6.306z^3 - 5.079z^2 + 2.026z - 0.3062} \quad (3)$$

There is an additional zero when getting the discrete transfer function from the continuous one. To design the controller blocks $\frac{S(z)}{R(z)}, \frac{T(z)}{R(z)}$ to fulfill the required response, follow the procedure given in ref. [7, 8]. The results are as follows:

$$\frac{S(z)}{R(z)} = \frac{-0.0422z^4 + 0.0793z^3 - 0.0200z^2 - 0.0262z + 0.0092}{z^4 - 0.4192z^3 - 0.6810z^2 + 0.0901z + 0.0118} \quad (4)$$

$$\frac{T(z)}{R(z)} = \frac{z^4 + 0.1971z^3 - 0.2922z^2 + 0.0429z}{z^4 - 0.4192z^3 - 0.6810z^2 + 0.0901z + 0.0118}$$

Then both the linear and nonlinear discrete-time responses for the designed MRAC can be obtained (Figs. 19 and 20). The results from this control technique were very good, a pure critical damped nonlinear response simulating the linear one even for the maximum ROC that can be required from the a/c (50 m/sec). The MRAC technique showed its power in not only stabilizing such poor systems but also achieving good time response characteristics. So, it is used along the rest of this paper in designing the control systems for the altitude and velocity hold autopilots.

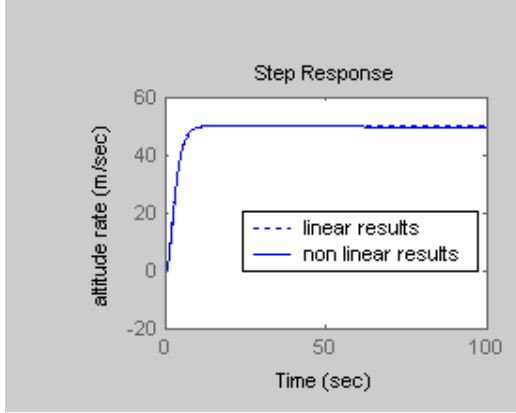


Fig. 19 ROC response to 50 m/sec ROC command using MRAC technique

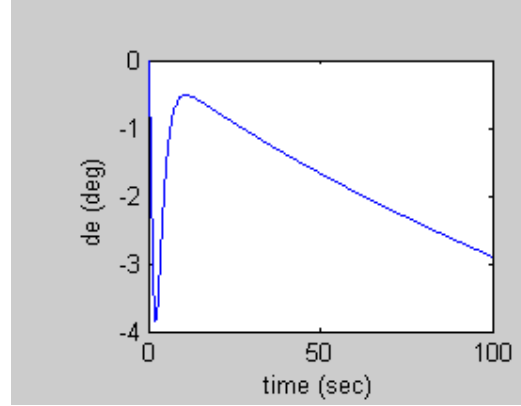


Fig. 20 Elevator deflection to 50 m/sec ROC command using MRAC technique.

3. Altitude Hold Autopilot

The ROC autopilot for every flight condition should be disengaged after reaching the required altitude. Then an altitude hold autopilot will be engaged, taking the responsibility for keeping stability at constant altitude, and resisting disturbances on altitude coming from gusts and velocity commands.

The block diagram of an MRAC for the altitude hold autopilot is shown in Fig. 21. The command input Δh_{comm} represents the altitude change command from the reference condition. The flight condition (2), 265 m/sec velocity and 10 km altitude [1], is taken as an example in designing the altitude hold autopilot. At this flight condition, the a/c is usually required to speed up at constant altitude. The open loop continuous and discrete-time transfer functions are given in equation (5), $T = 0.15$ sec.

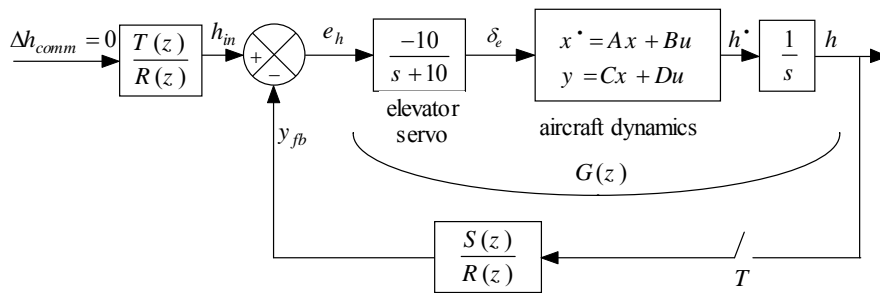


Fig. 21 Digital altitude-hold autopilot using MRAC

$$\frac{\Delta h(s)}{\Delta e_h(s)} = \frac{1}{s} \frac{\Delta h^*(s)}{\Delta e_h^*(s)} = \frac{1}{s} \frac{-151.4s^3 + 46.47s^2 + 1.536e004s + 37.68}{s^5 + 11.31s^4 + 21.27s^3 + 82.13s^2 + 1.133s + 0.6863} \quad (5)$$

$$\frac{\Delta h(z)}{\Delta e_h(z)} = \frac{-0.5014z^4 + 3.005z^3 - 1.061z^2 - 1.832z + 0.3934}{z^6 - 4.8799z^5 + 9.8316z^4 - 10.3987z^3 + 6.0054z^2 - 1.7419z + 0.1834}$$

The design criterion used in this section is not strictly specified as in the ROC autopilot. It is specified by means of iteration to get an acceptable nonlinear response; steady state error less than 10 m in altitude and T_s less than 50 sec.

To design the controller blocks $\frac{S(z)}{R(z)}$, and $\frac{T(z)}{R(z)}$ to fulfill the required response, follow the procedure given in ref. [7, 8].

$$\frac{S(z)}{R(z)} = \frac{2.2269z^5 - 9.2179z^4 + 14.9690z^3 - 11.7811z^2 + 4.3940z - 0.5916}{z^5 - 1.0732z^4 + 0.2298z^3 - 0.2032z^2 + 0.0275z + 0.0193} \quad (6)$$

$$\frac{T(z)}{R(z)} = \frac{z^5 - 0.4611z^4 - 0.0295z^3 + 0.0233z^2}{z^5 - 1.0732z^4 + 0.2298z^3 - 0.2032z^2 + 0.0275z + 0.0193}$$

The nonlinear discrete-time response for the designed MRAC can be tested by pushing the throttle lever forwardly to increase the a/c speed, and asking the altitude hold autopilot for constant altitude. The flight conditions (2), and (3) are at constant altitude but variable speeds (265 m/sec for flight. condition 2 and 350 m/sec for flight condition 3), so the altitude hold autopilot of flight condition (2) will operate until the a/c speed reaches the value of flight condition (3), at which the autopilot of flight condition (2) is disengaged and the autopilot of flight condition (3) is engaged. The change in throttle position required for the a/c to speed from 265 to 350 m/sec is below 5% of the throttle lever stroke (determined from the trim analysis), so to test the designed MRAC, 5% throttle lever increase is to be applied and constant altitude is to be required from the altitude hold autopilot. The results are shown in Figs. 22, 23, 24, and 25. The results from the MRAC altitude hold autopilot is satisfactory. Fortunately, the a/c speed response enhanced much more than that of the open loop one, although the velocity autopilot has not been engaged yet.

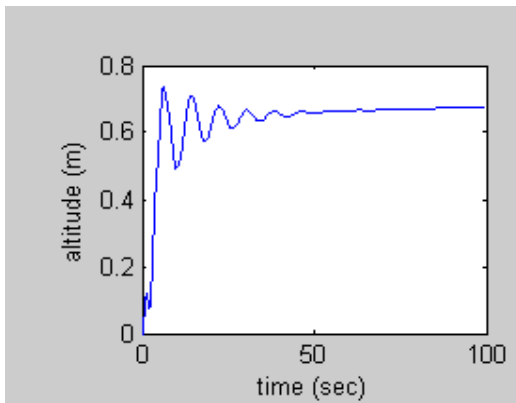


Fig. 22 Altitude response for 5% throttle lever increase while operating the MRAC altitude hold autopilot

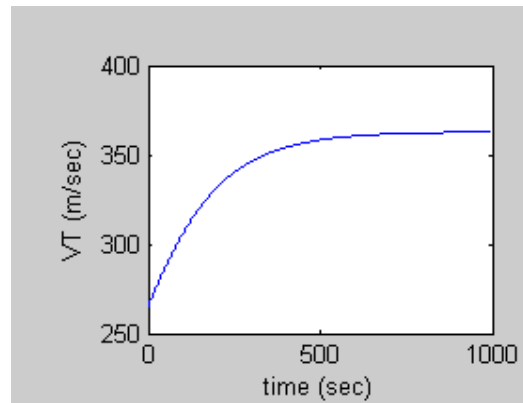


Fig. 23 A/C speed response for 5% throttle lever increase while operating the MRAC altitude hold autopilot

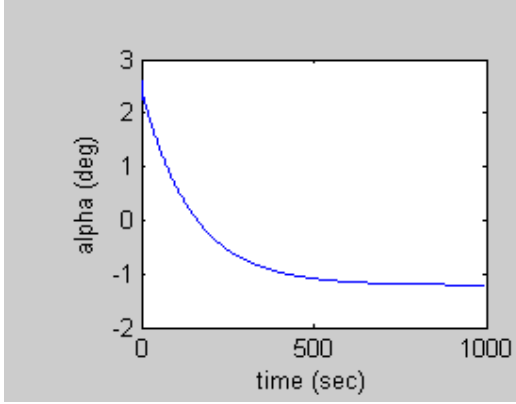


Fig. 24 Angle of attack response for 5% throttle lever increase while Operating the MRAC altitude hold autopilot

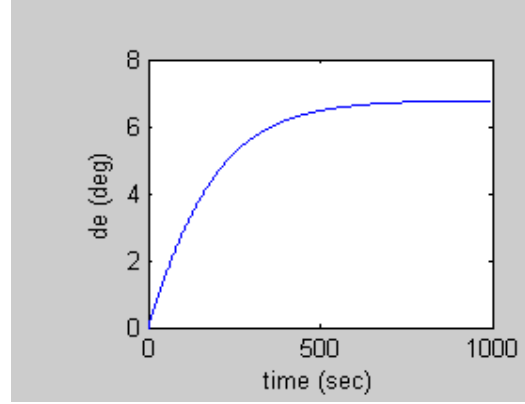


Fig. 25 Elevator deflection required for altitude hold with 5% throttle lever increase

4. Velocity Autopilot

The block diagram of an MRAC for the velocity autopilot is shown in Fig. 26. The flight condition (2) will be taken as an example in designing the velocity autopilot. Include the throttle servo and the engine time lag [2] with the a/c dynamics, the open loop transfer function is given by:

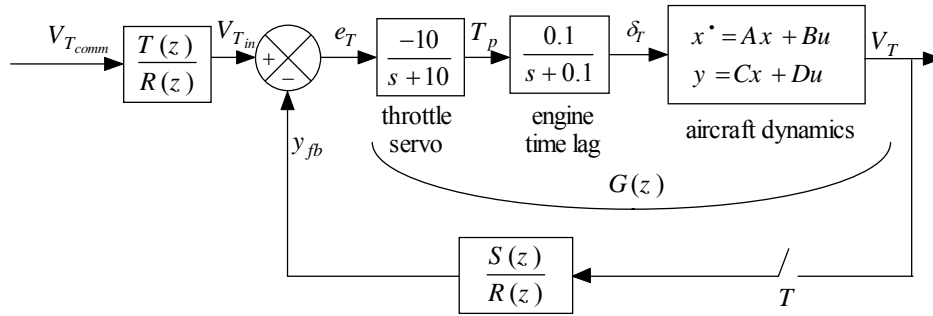


Fig. 26 Digital velocity autopilot using MRAC

$$\frac{\Delta V_T(s)}{\Delta e_T(s)} = \frac{-5.767s^3 - 3.97s^2 - 51.2s - 0.02969}{s^6 + 11.41s^5 + 22.4s^4 + 84.26s^3 + 9.346s^2 + 0.7996s + 0.06863} \quad (7)$$

$K_{dc} = -0.4326 \frac{m/sec}{volts}$ Where 1 volts corresponds to a unity change in $\Delta \delta_T$, while a unity change in $\Delta \delta_T$ corresponds to 0.65 % change in the throttle position. Note the -ve dc gain which means that an increase in the throttle lever position will lead to a decrease in the a/c velocity. This is because increasing the engine throttle of course increases the engine thrust, which in turn increases the a/c velocity at first. But the initial velocity increase leads to a +ve pitching moment through the term (C_{m_u}), which makes the a/c climb by a certain angle. In the climbing condition, the increased thrust is opposed by not only the drag as in the cruising case

but also a component of the a/c weight, hence the a/c velocity decreases below the initial value. This decrease in the a/c velocity leads to a -ve pitching moment, but the angle of attack decrease in climbing results in a +ve pitching moment, which compensates for the -ve pitching moment due to the velocity decrease. Thus, climbing results with a velocity lower than the cruising one. Most a/c's have its thrust-line below the c.g., i.e., increasing the engine throttle increases not only the engine thrust but also the +ve pitching moment on the a/c; supporting the previous behavior and leading to more -ve dc gain.

The response of the open loop transfer function is very bad due to the very poor zero (near the origin) located at -0.00058, which also led to a great difference between the linear and nonlinear responses. The open loop response to a 5 m/sec delta velocity command, which comes from $5/Kdc \Delta e_T$ command (about 7.5 % throttle decrease), are shown in Fig 27 and 28.

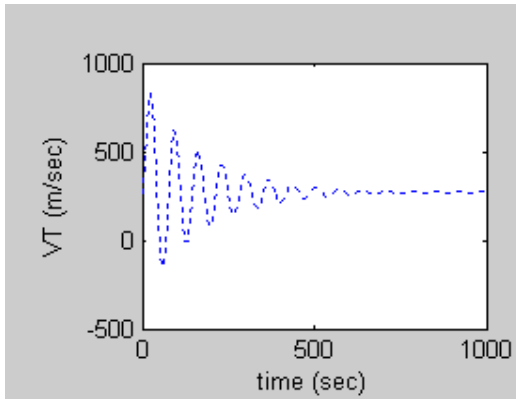


Fig. 27 a/c speed open loop linear response to 5 m/s delta velocity command

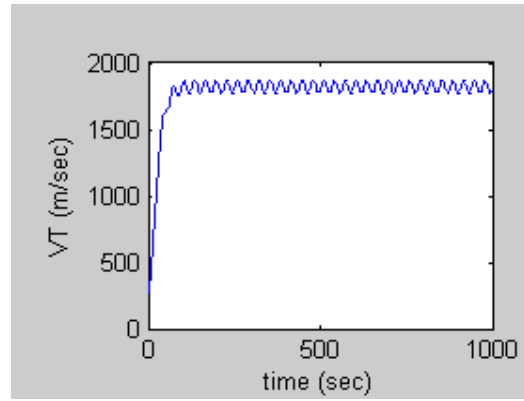


Fig. 28 a/c speed open loop non-linear response to 5 m/sec delta velocity command

4.1. Velocity MRAS Autopilot

To enhance this response, the MRAC technique (Fig. 26) is to be used due to its power in controlling such very poor systems. Discretizing the transfer function with an appropriate sampling period ($T_{samp} = 0.25 \text{ sec}$), the discrete-time transfer function will be:

$$\frac{\Delta V_T(z)}{\Delta e_T(z)} = \frac{-0.008512z^5 + 0.0004023z^4 + 0.02527z^3 - 0.0306z^2 + 0.0115z + 0.001928}{z^6 - 4.359z^5 + 7.897z^4 - 7.584z^3 + 3.97z^2 - 0.9824z + 0.05774} \quad (8)$$

At this flight condition (flight condition 2), due to the very poor dynamics and the high nonlinearity, the a/c velocity cannot be increased, using linear control techniques, from the trim value (265 m/sec) through one step command to the trim velocity of flight condition 3 (350 m/sec), i.e., 85 m/sec velocity increase command. Scheduling technique must be used to reach 350 m/sec velocity through two steps at minimum, and design two chronological cascaded linear controllers to accomplish this. To design the first controller, two criteria have to be specified; the settling time limit to be accepted from the controller for a critically damped response and the maximum velocity-increase command-input that will be demanded from the controller. These values were surveyed for such types of a/c's to be 100 ft/sec (say 35

m/sec) in about 25 sec settling time. To design the controller blocks $\frac{S(z)}{R(z)}$, and $\frac{T(z)}{R(z)}$ to fulfill the required response, follow the procedure stated previously:

$$\frac{S(z)}{R(z)} = \frac{-1.9221z^5 + 7.7889z^4 - 13.1803z^3 + 10.0152z^2 - 2.1981z - 0.5078}{z^5 - 2.3954z^4 + 2.2717z^3 - 0.8830z^2 - 0.0102z + 0.0170} \quad (9)$$

$$\frac{T(z)}{R(z)} = \frac{z^5 - 1.6769z^4 + 0.7683z^3 - 0.0401z^2 - 0.02z}{z^5 - 2.3954z^4 + 2.2717z^3 - 0.8830z^2 - 0.0102z + 0.0170}$$

The results for 35 m/sec velocity increase command are shown in Fig 29 and 30. The response was enhanced much but the settling time did not reach 25 sec yet and the overshoot exceeded 17 %. This is because of the very poor dynamics and the high nonlinearity at this flight condition. So to fulfill the desired response, online estimation, [7], is to be used for the transfer function parameters to update the controller parameters along the simulation time.

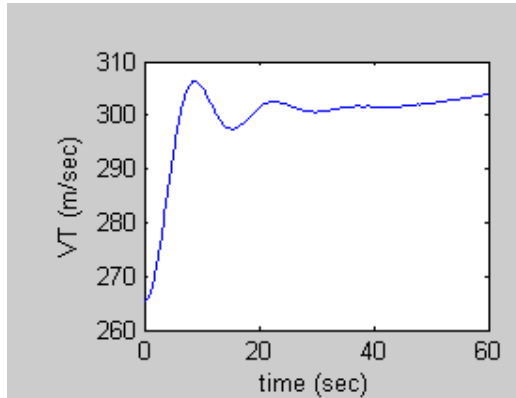


Fig. 29 A/C speed response to 35 m/sec velocity increase command

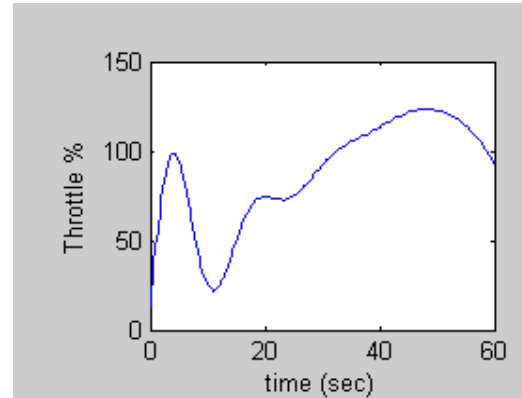


Fig. 30 Thrust required to 35 m/sec velocity increase command

4.2. Velocity MRAC Autopilot with Online Parameter Estimation and Controller Update:

The estimation algorithm and the online controller design scheme were incorporated into the control program. The online controller design scheme should be done carefully, because the transfer function may differ much from the original one. The controller parameters are not fed to the control program every sampling time as done when calculating the transfer function estimated parameters; this is for the controller to operate smoothly. So the problem concerning the frequency of feeding the newly designed controller parameters to the control program is solved by means of iteration. And it is found that 2 sec is an appropriate value for such application, i.e., the controller parameters are to be fed to the control program every 2 seconds. The responses of the previously designed MRAC velocity autopilot with engagement the parameter estimation, and controller update schemes are shown in Figs. 31, 32, and 33. A 35 m/sec velocity increase command was given to the system.

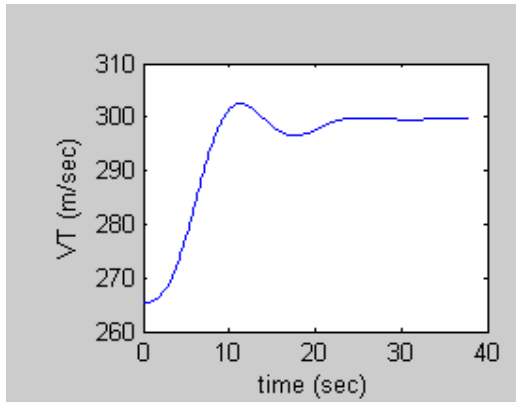


Fig. 31 A/C speed response to 35 m/sec velocity increase command, MRAC velocity autopilot with parameter estimation

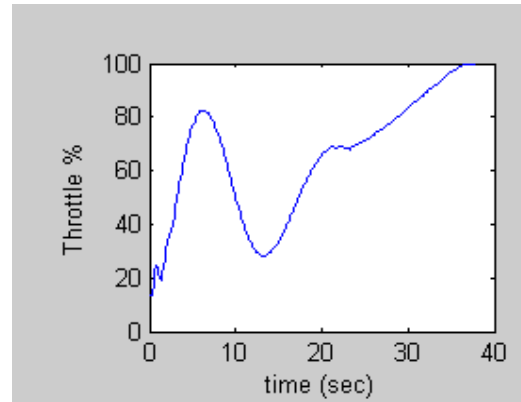


Fig. 32 Thrust required for 35 m/sec velocity increase command, MRAC velocity autopilot with parameter estimation

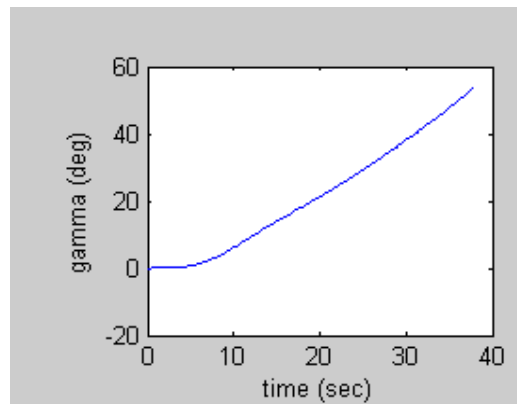


Fig. 33 Flight path angle response to 35 m/sec velocity increase command to the MRAC velocity autopilot with parameter estimation

The results are enhanced much. The thrust saturates due to the maintaining increase of the flight path angle which requires much thrust. This problem will vanish when engaging the altitude autopilot simultaneously with the velocity autopilot to hold the a/c altitude or command a certain ROC or flight path angle.

5. Conclusion

This paper presents appropriate control techniques for the ROC, altitude hold, and velocity hold autopilots of the FOXTROT a/c in certain flight conditions. For the ROC autopilot, both continuous and discrete time control techniques are used. The optimal control technique (Riccati) using state feedback was so far from achieving the required response characteristics. The pole placement technique using state feedback achieved better response than that achieved by the optimal control technique. But there was a poor zero, near the origin in the S-domain, that made the nonlinear response so peaky that it would be unstable if the ROC command increases above a certain limit, which is much lower than the maximum a/c ROC.

So, the technique which could handle this problem is the lag control; locating the controller pole to cancel the system poor zero, and the controller zero and gain to get the required response. This technique enhanced the response much and increased the nonlinear instability limit. But this limit could not exceed the maximum ROC of the a/c. Then the discrete time control technique (MRAC) is used to achieve this good response with increasing the ROC limit above the a/c maximum ROC. It is found that the MRAC technique is really appropriate in controlling such poor systems and reaching higher values for the ROC without instability occurs in the nonlinear response. Then it is used in controlling the altitude hold autopilot and gave good results. But in the velocity hold autopilot, the MRAC stand alone did not show the same capability for control as in the rate of climb and altitude hold autopilots; this is because the complexity of the velocity hold autopilot in some flight conditions. So an online system parameter estimation algorithm is used along with an online algorithm for MRAC design and implementation.

6. References

- [1] Donald McLean, "Automatic Flight Control Systems", Prentice Hall International Ltd, UK, 1990.
- [2] John H. Blakelock, "Automatic Control of Aircraft and Missiles", 2nd ed., John Wiley and Sons, Inc., 1991.
- [3] Jan Roskam, "
- [4] Dr. Robert C. Nelson, "Flight Stability and Automatic Control", 2nd ed., WCB/McGraw-Hill, 1998.
- [5] Brain L. Stevens, and Frank L. Lewis, "Aircraft Control and Simulation", John Wiley and Sons, Inc., 1992.
- [6] Katsuhiko Ogata, "Modern Control Engineering", 4th ed., Prentice Hall, New Jersey, 2002.
- [7] Karl John Åström, and Björn Wittenmark, "Adaptive Control", Addison-Wesley Publishing Company, 1989.
- [8] Katsuhiko Ogata, "Discrete – Time Control Systems", 2nd ed., Prentice Hall International, Inc., 1995.
- [9] Eric N. Johnson, "Adaptive Flight Control", Ph.D. Thesis, Georgia Institute of Technology, 2000.
- [10] Balbahadur Singh, "Online Aerodynamic Parameter Estimation for a Fault Tolerant Flight Control System", M.Sc. Thesis, Wichita State University, 2005.
- [11] Jung Soon Jang and Claire J. Tomlin, "Longitudinal Stability Augmentation System Design for the Dragon Fly UAV Using a Single GPS Receiver", AIAA, 2003-5592.
- [12] Vladislav Klein, "Application of System Identification to High Performance Aircraft", The George Washington University, Joint Institute for Advancement of Flight Sciences, NASA Langley Research Center, Hampton. Virginia 236111-0005.
- [13] Rolf T. Rysdyk and Anthony J. Calise, "Nonlinear Adaptive Flight Control using Neural Networks", Georgia Institute of Technology publications 2002.
- [14] Maruthi R, Akella, Kamesh Subbarao and John L. Junkins, "Non-Linear Adaptive Auto-Pilot for Uninhabited Aerial Combat Vehicles", Texas A & M University Publications, 2006.