Automatic Recovery and Autonomous Navigation of Disabled Aircraft After Control Surface Actuator Jam

Coşku Kasnakoğlu[1], Ünver Kaynak[2]

TOBB University of Economics and Technology, Ankara, 06560, Turkey

Loss of flight control authority in result of a system damage or an actuator jam is known or suspected to be the potential cause of many aviation accidents. After losing one or several control surface actuators, sustaining the flight for a safe return home depends on the capability of the remaining surfaces' control authority and the engine power. In this study, automatic recovery and autonomous guidance of a disabled general aviation aircraft is demonstrated. A nonlinear aircraft model and a MATLAB/SIMULINK based flight dynamics and control toolboxes are used to develop flight control laws for the disabled aircraft. The flight control laws are first validated for steady trimmed flight conditions and an actuator jam is applied to demonstrate the automatic recovery and autonomous navigation of the aircraft. The autopilot is designed to handle different control actuator malfunctions including rudder or aileron jams. Different scenario based simulations show that the new autopilot design is capable of sustaining safe flight and autonomous navigation under such malfunctions.

I. Introduction

In recent years, a number of commercial or general aviation accidents is known or suspected to be resulting from control surface damages or actuator jams[1]. Among these, especially the rudder malfunction is a prime suspect for some accidents[2]. The same phenomena are important for the military aircraft to survive after a battle damage of the lifting or control surfaces[3]. When an aircraft experiences a damage or loss of a control surface, the ensuing motion is called as an "unusual attitude" for the aircraft in which case the flight becomes "uncoordinated." This is because of the fact that remaining control surfaces should act to balance the abnormal forces and moments. In this case, the Euler angles may be in the opposite direction compared with those of the coordinated flight. Best example for this situation is the skidding or slipping of a healthy aircraft during crosswind landing. After a control surface actuator jam or some surface damage occurs, early or automatic recognition of the malfunction, sustaining the flight using the rest of the controls, and finally safe landing of the aircraft is of prime importance for survivability. In this case, it is very important that if certain aircraft can sustain its flight using the rest of the controls, and if so, within which envelope can this aircraft achieve this[4] becomes important. Besides, even if the remaining control power is sufficient, the human pilot workload may become excessive, and an autopilot is required. Under these conditions, the requirement for survivability is an autopilot design for these "unusual attitudes." The modern research work in this area is progressing in the direction of "intelligent control."[5] Most of the times, classical control algorithms do not prove to be sufficient and modern control techniques such as the optimal control, robust control, nonlinear control and quite recently the fuzzy, genetic or neural control techniques are frequently addressed[6].

[1] Assistant Professor, Department of Electrical and Electronics Engineering, Member AIAA
[2] Professor, Department of Mechanical Engineering , Member AIAA

II. Methodology

A. Problem Statement

In this study, an autopilot will be developed for a general aviation aircraft encompassing the necessary flight control laws against unusual attitudes. The autopilot will be capable of executing an automatic recovery after a rudder of aileron actuator jam, followed by a steady maneuvering flight to join an approach path for a local airport, and finally demonstrating an automatic landing under variable crosswind conditions. For this, the freely available computer simulation and control software called as the "FDC 1.2 – A SIMULINK Toolbox for Flight Dynamics and Control Analysis"[7] will be used to develop the flight control laws. The nonlinear dynamic model of the Cessna 172 aircraft

Figure 1. Cessna 172 aircraft. (Photo courtesy of P. Alejandro Diaz)

will be used in the design, development and test of the flight control laws under different scenarios. Certainly, the dynamic model of the aircraft must change in parallel with the unusual attitudes. However, development of the true dynamic model requires rather costly wind tunnel testing including use of rotary balances, but there are some low cost remedies including the use of artificial neural network technique[8] for deriving the stability derivatives. Nevertheless, the dynamic model is assumed to be constant in this study, provided that the amount of deviation from usual angles will be kept small.

B. Flight Dynamics of Unusual Attitudes

Nonlinear dynamic model is used to accommodate the uncoordinated motion of the aircraft following the loss of a control surface. As known, normal operation of a rudder and ailerons are in coordination (acting in the same sense) with each other in usual flight, but they must operate in opposite directions for unusual attitudes. Figure 1 gives the commonly used sign convention for the flight control surfaces for an aircraft[7]. For instance, in order to make a coordinated turn to the left, the control wheel is turned to the left and a left rudder pressure is applied. In order to accommodate the left turn command, right aileron moves down (+), left aileron moves up (-) and the rudder moves left (-). This causes the aircraft to bank to the left and the nose travels to the left in a coordinated action. On the contrary, in case of a rudder malfunction in which the servoactuator locks to the left (+), opposite aileron travel is necessary where the right aileron moves up (-) and the left aileron moves down (+) for counteracting the left turning tendency. The aircraft is now put in an "usual attitude" where the nose points to the left, but the aircraft banks to the right. During this motion, appropriate level of throttle and elevator commands are necessary for the complete recovery and obtaining trim values. Small values of rudder lock may be relatively easy to accomodate with opposite aileron movements, but if the rudder lock angles get bigger, the remaining control authority with other controls and engine power will have upper bounds based on the specific aircraft design.

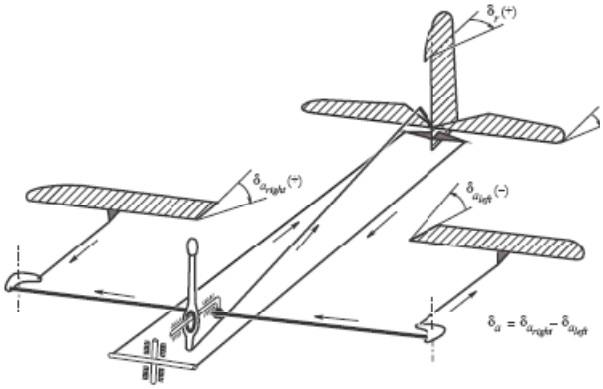

Figure 2. Sign convention for flight control surfaces [7].

C. Nonlinear Aircraft Model

Nonlinear aircraft model in a state space form can be written in body, stability or wind axes as given in Equation 1 where $\boldsymbol{x}$ is the state vector, and $\boldsymbol{F}_{tot}$ and $\boldsymbol{M}_{tot}$ are the total external forces and moments respectively.

$$\dot{x} = f(x, F_{tot}(t), M_{tot}(t)) \tag{1}$$

The state vector in wind axes is given by[7]

$$x = \left[V, \alpha, \beta, p, q, r, \varphi, \theta, \psi, x_e, y_e, h\right]^T \tag{2}$$

where V is the total velocity, α and β are the angle of attack and angle of sideslip, p, q, r, are the body angular velocities, φ, θ, ψ are the body Euler angles, and x_e, y_e, h, are the position coordinates with respect to the Earth fixed noninertial axes. Total external force and moment vectors under the effects of aerodynamic, propulsive, gravity and athmospheric forces are given as.

$$F_{tot} = f_F(x(t), u(t), v(t), t) \tag{3}$$

$$M_{tot} = f_M(x(t), u(t), v(t), t) \tag{4}$$

For insance, the aerodynamic force and the aerodynamic input vector are given by

$$F_{aero} = f_{aero}(x, \dot{x}, u_{aero}) \tag{5}$$

$$u_{aero} = \left[\delta_e, \delta_a, \delta_r, \delta_t\right]^T \tag{6}$$

where δ_e is the elevator angle, δ_a aileron angle, δ_r rudder angle and δ_t is the throttle travel. Finaly, the aerodynamic forces and moments in terms of nondimensional variables (stability derivatives) are given as[7]

$$C_{X_a} = C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\alpha^3}}\alpha^3$$

$$+C_{X_q}\frac{q\bar{c}}{V}+C_{X_{\delta_r}}\delta_r+C_{X_{\delta_f}}\delta_f+C_{X_{\alpha\delta_f}}\alpha\delta_f \tag{7}$$

$$C_{Y_a}=C_{Y_0}+C_{Y_\beta}\beta+C_{Y_p}\frac{pb}{2V}+C_{Y_r}\frac{rb}{2V}$$
$$+C_{Y_{\delta_a}}\delta_a+C_{Y_{\delta_r}}\delta_r+C_{Y_{\delta_r\alpha}}\delta_r\alpha+C_{Y_{\dot{\beta}}}\frac{\dot{\beta}b}{2V} \tag{8}$$

$$C_{Z_a}=C_{Z_0}+C_{Z_\alpha}\alpha+C_{Z_{\alpha^3}}\alpha^3+C_{Z_q}\frac{q\bar{c}}{V}$$
$$+C_{Z_{\delta_e}}\delta_e+C_{Z_{\delta_e\beta^2}}\delta_e\beta^2+C_{Z_{\delta_f}}\delta_f+C_{Z_{\alpha\delta_f}}\alpha\delta_f \tag{9}$$

$$C_{l_a}=C_{l_0}+C_{l_\beta}\beta+C_{l_p}\frac{pb}{2V}+C_{l_r}\frac{rb}{2V}$$
$$+C_{l_{\delta_a}}\delta_a+C_{l_{\delta_r}}\delta_r+C_{l_{\delta_a\alpha}}\delta_a\alpha \tag{10}$$

$$C_{m_a}=C_{m_0}+C_{m_\alpha}\alpha+C_{m_{\alpha^2}}\alpha^2+C_{m_q}\frac{q\bar{c}}{V}$$
$$+C_{m_{\delta_e}}\delta_e+C_{m_{\beta^2}}\beta^2+C_{m_r}\frac{rb}{2V}+C_{m_{\delta_f}}\delta_f \tag{11}$$

$$C_{n_a}=C_{n_0}+C_{n_\beta}\beta+C_{n_p}\frac{pb}{2V}+C_{n_r}\frac{rb}{2V}$$
$$+C_{n_{\delta_a}}\delta_a+C_{n_{\delta_r}}\delta_r+C_{n_q}\frac{q\bar{c}}{V}+C_{n_{\beta^3}}\beta^3 \tag{12}$$

D. Computer Model of the Aircraft

For the computer modeling of the aircraft, we utilize the Airlib library[15] for MATLAB/SIMULINK, which is based on dynamical model of the aircraft in the FDC Toolbox. The equations of motion are as described in the previous section, where the geometry, mass, and aerodynamic derivatives for the Cessna 172 aircraft are as follows:

Geometry and Mass Parameters:

$$[cbar\ \ b\ \ S\ \ Ix\ \ Iy\ \ Iz\ \ Jxy\ \ Jxz\ \ Jyz\ \ m] = [1.4935\ \ 10.9118\ \ 16.1651\ \ 1285.3\ \ 1824.9\ \ 2666.9\ \ 0\ 0\ 0\ \ 1043.3]$$

Aerodynamic D-Force Derivatives:

$$[CD0\ CDa\ CDq\ CDde\ CDih] = [0.031\ \ 0.13\ \ 0\ \ 0.06\ \ 0]$$

Aerodynamic L-Force Derivatives:

$$[CL0\ CLa\ CLq\ CLde\ CLih] = [0.31\ \ 5.143\ \ 3.9\ \ 0.43\ \ 0]$$

Aerodynamic Y-Force Derivatives:

$$[CY0\ CYb\ CYp\ CYr\ CYda\ CYdr] = [0\ \ -0.31\ \ -0.037\ \ 0.21\ \ 0.0\ \ 0.187]$$

Aerodynamic X-moment Derivatives:

$$[Cl0\ Clb\ Clp\ Clr\ Clda\ Cldr] = [0\ \ -0.089\ \ -0.47\ \ 0.096\ \ -0.178\ \ 0.0147]$$

Aerodynamic Y-Moment Derivatives:

$$[Cm0\ Cma\ Cmq\ Cmde\ Cmih] = [-0.015\ \ -0.89\ \ -12.4\ \ -1.28\ \ 0]$$

Aerodynamic Z-moment Derivatives:

$$[Cn0\ Cnb\ Cnp\ Cnr\ Cnda\ Cndr] = [0 \quad 0.065 \quad -0.03 \quad -0.099 \quad -0.053 \quad -0.0657]$$

The International Measurement System (MKS) is adopted for all the units.

III. Autopilot Design

A. Straight and Level Flight

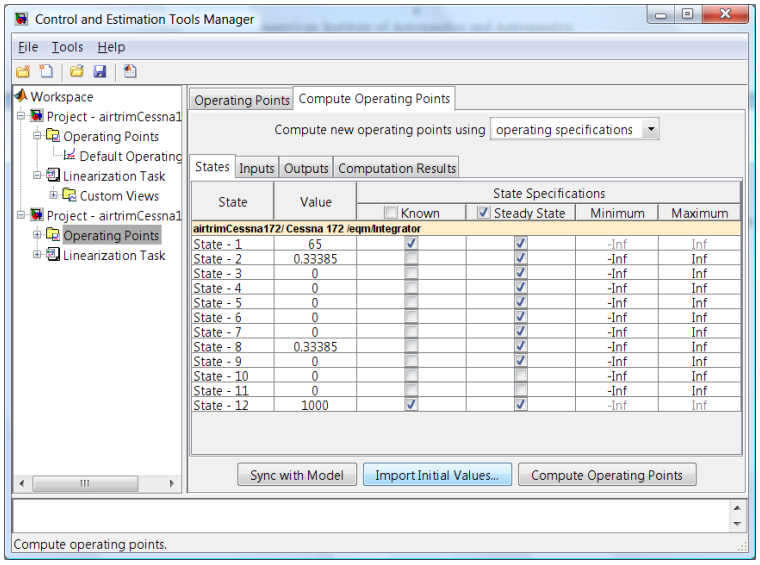

Figure 3. Computing an operating point using Control Design and Estimation Tools Manager in MATLAB.

We start with an autopilot design when the aircraft is flying at a cruise speed of 65 m/s, at an altitude of 1000 m and all control surface servoactuators are functioning properly. The aircraft is first trimmed using the optimization routines under the Control and Estimation Tools Manager of MATLAB (see Figure 3), and using the functions of the Airlib library. The trim points satisfying the below given operating conditions

$$v = 65\ \text{m/s}, \qquad z_e = 1000\ \text{m}, \qquad \frac{\partial\alpha}{\partial t} = \frac{\partial\beta}{\partial t} = \frac{\partial p}{\partial t} = \frac{\partial q}{\partial t} = \frac{\partial r}{\partial t} = \frac{\partial\psi}{\partial t} = \frac{\partial\theta}{\partial t} = \frac{\partial\phi}{\partial t} = \frac{\partial z_e}{\partial t} = 0 \tag{13}$$

are computed as follows:

$$\begin{aligned} x_0 &= [v, \alpha, \beta, p, q, r, \psi, \theta, \phi, x_e, y_e, z_e] = [65\ , -0.0072915, 0, 0, 0, 0, 0, -0.0072915, 0, 0, 0, 1000] \\ u_0 &= [F_x, \delta_e, \delta_a, \delta_r] = [1125.7, -0.0066489, 0, 0] \end{aligned} \tag{14}$$

where x_o is the aircraft state at the trim condition and u_0 is the input vector containing the throttle and surface deflections to be applied. The nonlinear aircraft with an initial state x_0, its inputs u_0 and its parameters as given in Section II is simulated by using SIMULINK and the results are plotted in Figures 4-5. It can be seen from the figures that the aircraft maintains straight and level flight. Apart from the twelve state parameters, two additional parameters that are important in autopilot design, namely $A_y\ (g)$ and $\chi\ (rad)$ are also plotted. The former parameter A_y is the acceleration in direction perpendicular to the aircraft heading. It is desirable that this acceleration be small for flight comfort as well as to achieve the turn coordination (i.e. keep slip or skid small during

turns). The latter parameter, which is the azimuth angle $\chi = \psi + \beta$, is the actual direction in which the aircraft is flying. This is normally the same as the heading angle ψ since β is zero during a normal flight. However, for the autopilot designs in the succeeding sections, the aircraft will be flying at unusual attitudes and hence it will be the case that $\beta \neq 0$. In these cases, it is important to be able to control the actual direction of motion (χ) as opposed to which way the nose of the aircraft is pointed (ψ).

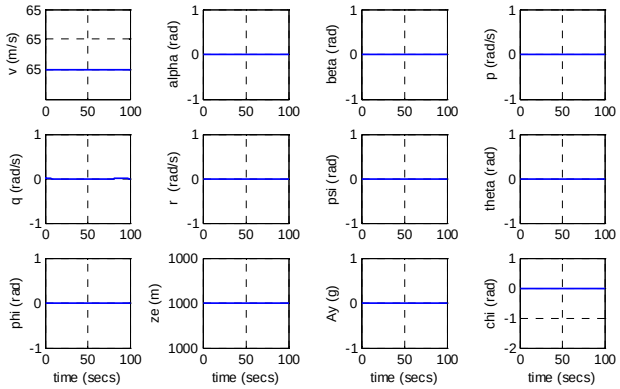

Figure 4. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model for normal flight around $v = 65\ m/s$ and $h = 1000\ m$.

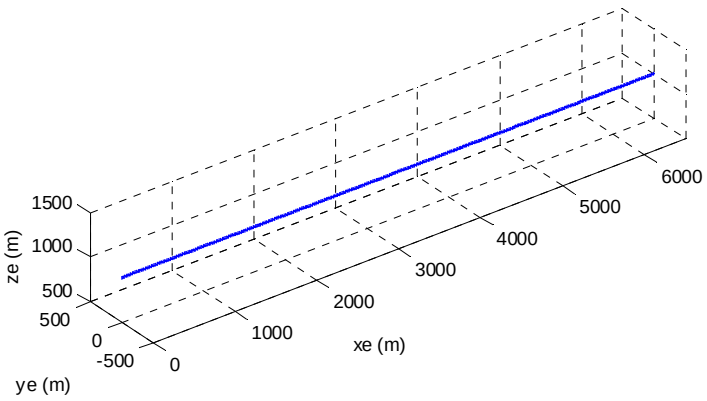

Figure 5. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model for normal flight around $v = 65\ m/s$ and $h = 1000\ m$.

The next task is to design an autopilot for normal flight that will carry out a set of desired commands from the operator. The types of commands that the autopilot will receive from the operator are the airspeed (v_{cmd}), altitude ($z_{e,cmd}$), direction of travel (χ_{cmd}) and the lateral acceleration command ($A_{y,cmd}$). The last command will normally be set to zero to minimize the lateral force acting on the aircraft. Since a change in the direction of travel (χ) is achieved by the banking of the aircraft (i.e. a change in ϕ) it usually leads better results to first design an inner controller to set the roll angle (ϕ) to a desired value, and then wrap an outer controller to send the command ϕ_{cmd} to this inner controller based on the direction of travel command χ_{cmd} as in Figure 6.

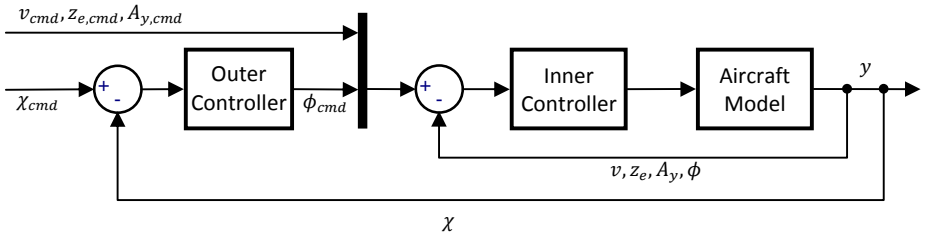

Figure 6. Closed loop system structure for autopilot design.

For inner controller design we first linearize the nonlinear aircraft model about the trim conditions (x_0, u_0), which yields a linear state-space system G of the form:

$$G \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{15}$$

where the output vector y consists of the signals that we wish to control, namely $y = [v, z_e, \phi, A_y]$. The open loop step response for the linearized system (15) is shown in Figure 7.

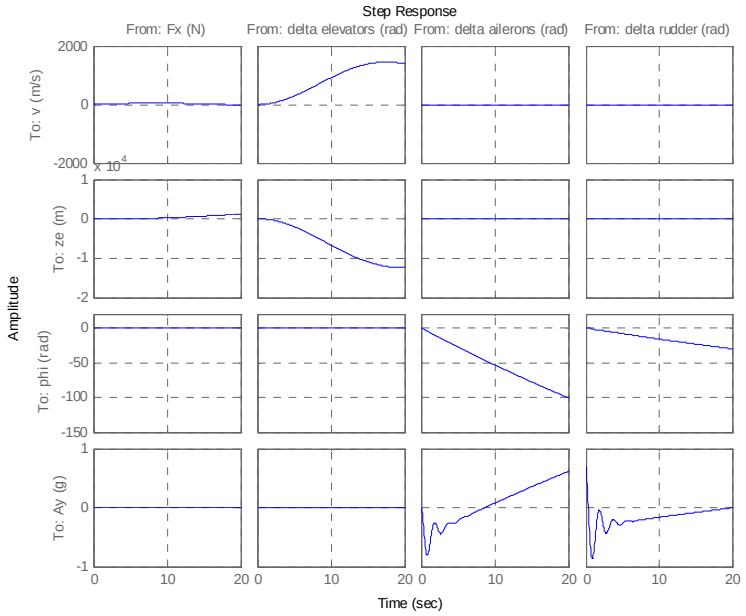

Figure 7. Aircraft states resulting from simulations of the linearized aircraft model in open-loop for normal flight around $v = 65\ m/s$ and $h = 1000\ m$.

The goal is to design a controller for this system so that the closed loop system tracks a desired reference $(v_{cmd}, z_{e,cmd}, \phi_{cmd}, \chi_{cmd})$. The typical approach in controller design for aircrafts is to design two separate controllers for the longitudinal and lateral directions and then use these controllers as if the two dynamics were uncoupled. While this approach is often preferred for its simplicity, in real life there is usually significant coupling

between longitudinal and lateral directions which creates undesired errors when the two controllers are used together. In this paper we utilize a design approach, namely *loop-shaping*, where we can explicitly address and minimize the coupling effects between different channels. To briefly summarize, the loop-shaping method computes a stabilizing H_∞ controller K for the plant G to shape the sigma (σ) plot of the transfer function GK to have a desired loop shape G_d, within a certain accuracy γ. Ideally γ should be as small as possible. The procedure for determining K is as follows: First the GCD formula described in Ref.12 is used to compute a stable minimum-phase loop-shaping squaring-down prefilter W so that the shaped plant $G_s = GW$ is square and the desired loop shape G_d is approximated as closely as possible. That is

$$\sigma(G_d) \approx \sigma(G_s) \quad \forall \omega \in [\omega_{\min}, \omega_{\max}] \tag{16}$$

where σ denotes singular values and $[\omega_{\min}, \omega_{\max}]$ is the desired frequency range. This step is followed by the computation of an optimal loop-shaping controller K_s for the shaped plant G_s using the normalized-coprime-factor control synthesis theory [13]. The overall controller is then given by

$$K = WK_s \tag{17}$$

If the plant G is a continuous time LTI, the D matrix of G is full rank, and G has no finite zeros on the $j\omega$-axis, and $[\omega_{\min}, \omega_{\max}] = [0, \infty]$ then a perfect fit $\sigma(G_d) = \sigma(G_s)$ can theoretically be achieved for all frequencies. Otherwise the range $[\omega_{\min}, \omega_{\max}]$ can be used to achieve a pole-shifting bilinear transform [14] to compute a shifted system G_{shift} to which the loop-shaping procedure above is applied. This results in an approximate fit for the original plant G over the desired frequency range. The reader interested in further details of the loop-shaping procedure is referred to Refs.12-14.

For inner controller design task at hand, we set the desired loop shape G_d as follows:

$$G_d(s) = \text{diag}\left(\frac{2}{s}, \frac{2}{s}, \frac{2}{s}, \frac{2}{s}\right) = \begin{bmatrix} \frac{2}{s} & 0 & 0 & 0 \\ 0 & \frac{2}{s} & 0 & 0 \\ 0 & 0 & \frac{2}{s} & 0 \\ 0 & 0 & 0 & \frac{2}{s} \end{bmatrix} \tag{18}$$

where diag stands for diagonal matrix. This means the closed loop transfer function matrix $T(s)$ from the references $(v_{cmd}, z_{e,cmd}, A_{y,cmd}, \phi_{cmd})$ to outputs (v, z_e, A_y, ϕ) will approximately be the following

$$T(s) = G_d(s)[I + G_d(s)]^{-1} \approx \text{diag}\left(\frac{2}{s+2}, \frac{2}{s+2}, \frac{2}{s+2}, \frac{2}{s+2}\right) = \begin{bmatrix} \frac{2}{s+2} & 0 & 0 & 0 \\ 0 & \frac{2}{s+2} & 0 & 0 \\ 0 & 0 & \frac{2}{s+2} & 0 \\ 0 & 0 & 0 & \frac{2}{s+2} \end{bmatrix} \tag{19}$$

Observe that:

1. The individual transfer functions for the diagonal channels are approximated by

$$\frac{V_{cmd}(s)}{V(s)} = \frac{z_{e,cmd}(s)}{z_e(s)} = \frac{A_{y,cmd}(s)}{A_y(s)} = \frac{\phi_{cmd}(s)}{\phi(s)} \approx \frac{2}{s+2} \tag{20}$$

This means the closed loop system will be able to track all references successfully, with a settling time of approximately $t_s = 5\tau = 5\frac{1}{2} = 2.5$ seconds.

2. The off-diagonal entries of $T(s)$ are approximately zero. This means that the coupling between different command-response pairs are eliminated; e.g. when the operator issues a step input from v_cmd, this command will result in an increase of v by 1 m/s, but this command will not result in any other uninented effects such as a change in altitude, lateral accelaration or roll angle. This point is sometimes neglected in simple autopilot designs but needs to be addressed explicitly in this paper, especially since in the succeeding sections we will be dealing with unusual flight attitudes, which introduce a fair amount coupling between individual channels.

The validity of the items above can be confirmed by looking at the closed loop step response of the system, which is shown in Figure 8. As expected, the plots in the diagonals converge to the value 1 in about 2.5 seconds and the off-diagonal plots are virtually identical to zero.

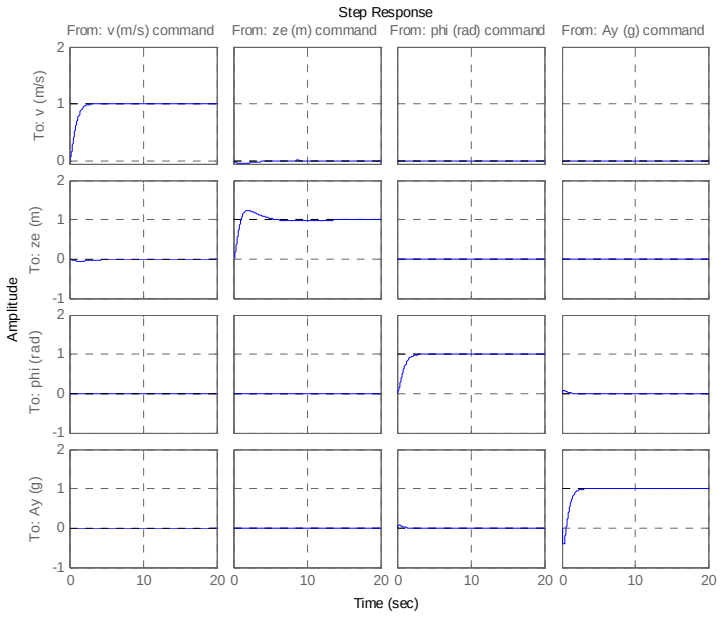

Figure 8. Aircraft states resulting from simulations of the linearized aircraft model in closed-loop for normal flight around $v = 65\ m/s$ and $h = 1000\ m$.

The final step is to test the autopilot designed in closed loop with the nonlinear aircraft model from Airlib and perform SIMULINK simulations. We must first wrap the controller designed above with an outer controller (see Figure 6) which will generate an appropriate roll angle command ϕ_{cmd} from a desired direction of motion χ_{cmd}. Since this is a single input single output (SISO) controller, we utilize a simple proportional integral derivative (PID) controller with filtered derivative (to yield a practically realizable system and also to reduce the effects of noise associated with differentiation) of the following form

$$\frac{\Phi_{cmd}(s)}{\mathrm{X}_{cmd}(s)} = K_p + \frac{K_i}{s} + \frac{K_d s}{s/N + 1} \tag{21}$$

where the coefficients were selected empirically as $K_p = 4$, $K_i = 1$, $K_d = 1$ and $N = 100$. This outer controller, the inner controller designed above and the nonlinear Cessna 172 model from Airlib were connected as illustrated in

Figure 6 and the resulting system was simulated in SIMULINK, the results of which are presented in Figure 9 – Figure 10. The scenario studied in this simulation is as follows: 1) The autopilot is commanded to maintain level flight at $65\,m/s$ and $1000\,m$ from $t = 0s$ to $t = 10\,s$. 2) The autopilot is commanded to increase the aircraft's altitude to $1250\,m$ between $t = 10\,s$ and $t = 30\,s$. 3) The autopilot is commanded to decrease the aircraft's speed to 45 m/s between $t = 30\,s$ and $t = 50\,s$. 4) The autopilot is instructed to perform a $90°$ turn between $t = 60\,s$ and $t = 80\,s$. 5) The autopilot is directed to simultaneously increase the airspeed to $65\,m/s$, increase the altitude to $1500\,m$ and perform a $45°$ turn between $t = 110\,s$ and $t = 140\,s$. In Figure 9, the commands applied to the autopilot are shown in green dotted lines, and the aircraft's response is in blue solid lines, from where it can be seen that desired references are followed quite closely. The trajectory of the aircraft in 3D space is shown in Figure 10.

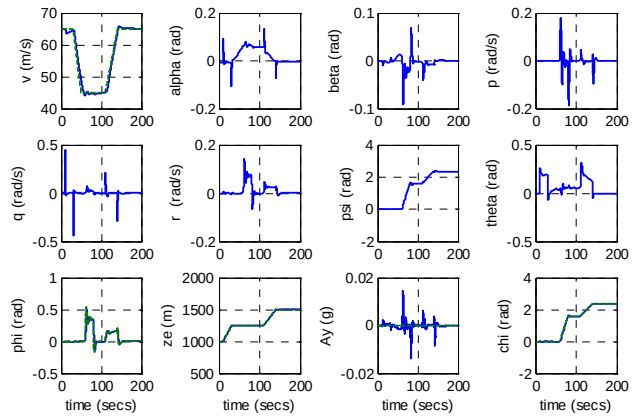

Figure 9. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under normal operation.

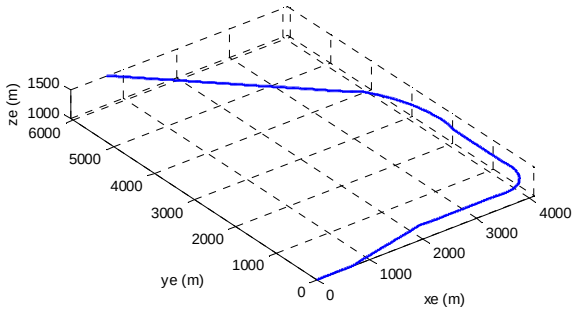

Figure 10. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under normal operation.

B. Rudder Servoactuator Jam

In this section and the next we consider autopilot designs for the cases when the servoactuators commanding the control surfaces are stuck. We first consider the case where the rudder servoactuator is jammed, and for the sake of example, at an angle of ten degrees, (i.e. $\delta_r = 10° = 0.1745\,rad$) while the aircraft is flying at a cruise speed of

$65\ m/s$ and an altitude of $1000\ m$. The uncontrolled response of the aircraft obtained from SIMULINK simulations using the nonlinear Cessna 172 model from Airlib is shown in Figures 11 and 12.

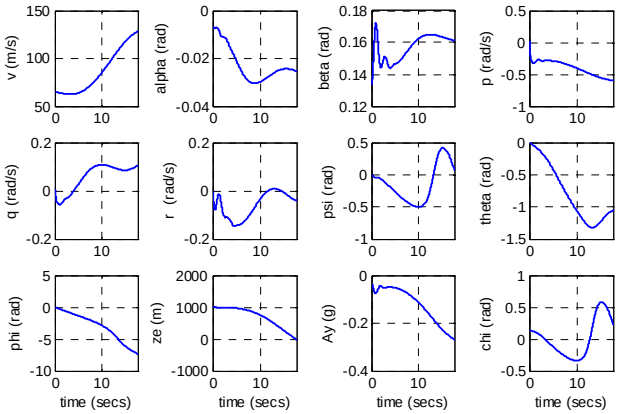

Figure 11. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model for rudder jam around $v = 65\ m/s$ and $h = 1000\ m$.

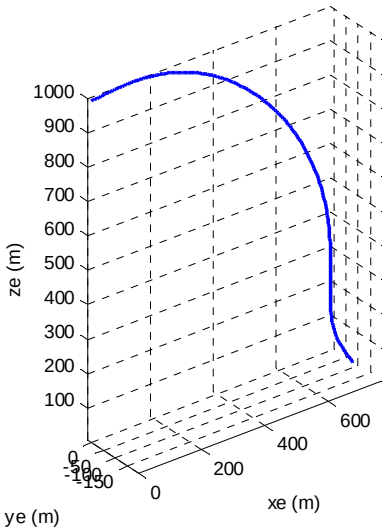

Figure 12. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model for rudder jam around $v = 65\ m/s$ and $h = 1000\ m$.

It can be seen from the figures that aircraft starts rolling around its axis, loses altitude and hits the ground at a high speed in less than 20 seconds. This is a relatively short amount of time for a human pilot to show an appropriate reaction to save the aircraft; hence the usefulness of an emergency autopilot to handle such a situation.

Similar to the previous section, the aircraft is first trimmed using the optimization routines under the Control and Estimation Tools Manager of MATLAB (see Figure 3) and using the functions of the Airlib library to compute trim points satisfying the operating specifications, which are as follows:

$$\mathrm{v} = 65\ \mathrm{m/s},\ \ \mathrm{z_e} = 1000\ \mathrm{m}, \delta_\mathrm{r} = 0.1745\ \mathrm{rad},\ \ \frac{\partial \alpha}{\partial \mathrm{t}} = \frac{\partial \beta}{\partial \mathrm{t}} = \frac{\partial \mathrm{p}}{\partial \mathrm{t}} = \frac{\partial \mathrm{q}}{\partial \mathrm{t}} = \frac{\partial \mathrm{r}}{\partial \mathrm{t}} = \frac{\partial \psi}{\partial \mathrm{t}} = \frac{\partial \theta}{\partial \mathrm{t}} = \frac{\partial \phi}{\partial \mathrm{t}} = \frac{\partial \mathrm{z_e}}{\partial \mathrm{t}} = 0 \qquad (22)$$

The trim point resulting for this specification is as follows:

$$\begin{aligned} x_0 &= [v, \alpha, \beta, p, q, r, \psi, \theta, \phi, x_e, y_e, z_e] \\ &= [65, -0.0073197, 0.13367, 0, 0, 0, 0, -0.0029238, 0.032667, 0, 0, 1000] \\ u_0 &= [F_x, \delta_e, \delta_a, \delta_r] = [1170.6, -0.0066292, -0.052421, 0.17453] \end{aligned} \qquad (23)$$

where x_o is the aircraft state at the trim condition and u_0 is the input vector containing the throttle and surface deflections to the applied. Comparing with the trim point above with that of the normal flight given in (14), one can make two observations: 1) To compensate for the rudder lock, it is necessary to increase the throttle (i.e. increase F_x) and also apply aileron input (δ_a) in the opposite direction. 2) To maintain the desired airspeed and altitude the aircraft needs to fly at an unusual attitude; note the non-zero values for the angles β and ϕ need, which implies that the aircraft is banked and sliding sideways. Since the rudder is unusable, we have only three control inputs (F_x, δ_e and δ_a) left, and hence it will only be possible to control three quantities for this situation (as opposed to four quantities in the previous section). We therefore sacrifice A_y (and therefore flight comfort and turn coordination) and set our outputs to be controlled as $y = [v, z_e, \chi]$. Note again that it is the actual direction of motion (χ) that must be controlled and not merely the heading (ψ) of the aircraft; this is because for flights at unusual attitudes we have a non-zero and non-constant sideslip angle β and thus $\chi = \psi + \beta \neq \psi$. The task is therefore to design an autopilot that will carry out velocity (v_{cmd}), altitude ($z_{e,cmd}$) and direction of travel (χ_{cmd}) commands. As before, we will first design an inner controller to set the roll angle (ϕ) to a desired value, and then wrap an outer controller to send the command ϕ_{cmd} to this inner controller based on the direction of travel command χ_{cmd}. The closed loop system structure is identical to that in Figure 6, except that the lateral acceleration A_y and its command $A_{y,cmd}$ are no longer part of the picture. As before, for inner controller design we first linearize the nonlinear aircraft model about the trim conditions (x_0, u_0), which yields a linear state-space system G of the form where the output vector y is $y = [v, z_e, \chi]$. The open loop step response for the linearized system is shown in Figure 13.

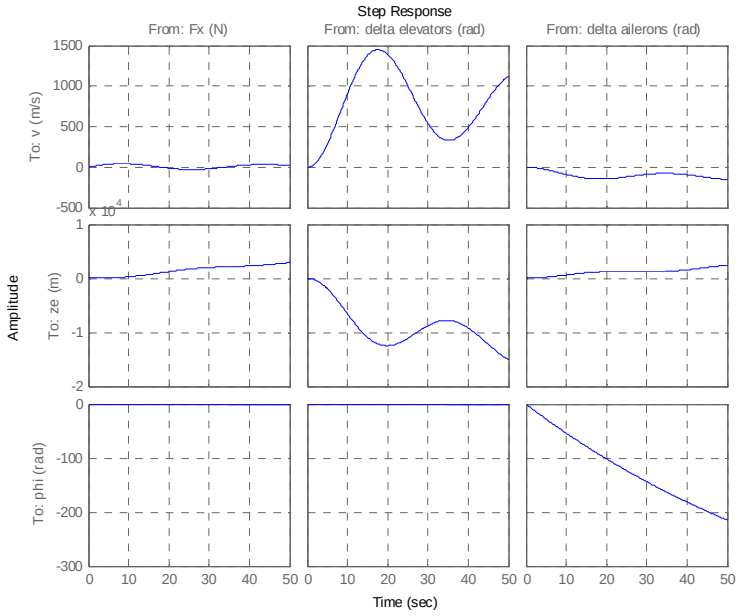

Figure 13. Aircraft states resulting from simulations of the linearized aircraft model in open-loop under rudder jam around $v = 65\ m/s$ and $h = 1000\ m$.

To design the inner controller we use the loop-shaping approach described in the previous section with a desired loop-shape

$$G_d(s) = \text{diag}\left(\frac{1}{s}, \frac{1}{s}, \frac{1}{s}\right) \tag{24}$$

The loop-shape is less aggressive compared to the normal flight scenario considered in the previous section ($1/s$ instead of $2/s$) since the aircraft is less stable at unusual attitudes and has a higher tendency to leave the vicinity of the trim point. The closed-loop transfer function for the inner loop resulting from this target loop-shape will approximately be

$$T(s) = G_d(s)[I + G_d(s)]^{-1} \approx \text{diag}\left(\frac{1}{s+1}, \frac{1}{s+1}, \frac{1}{s+1}, \frac{1}{s+1}\right) \tag{25}$$

Note that due to the less aggressive desired loop-shape G_d, the settling time is $t_s = 5\tau = 5\frac{1}{1} = 5$ seconds, which is twice that of the normal flight scenario. The closed loop step response of the system is shown in Figure 14. Note that the diagonal plots converge to one, which implies that the desired tracking is achieved, and the off-diagonal plots are close to zero, which implies that the coupling effects between channels is negligible in closed-loop.

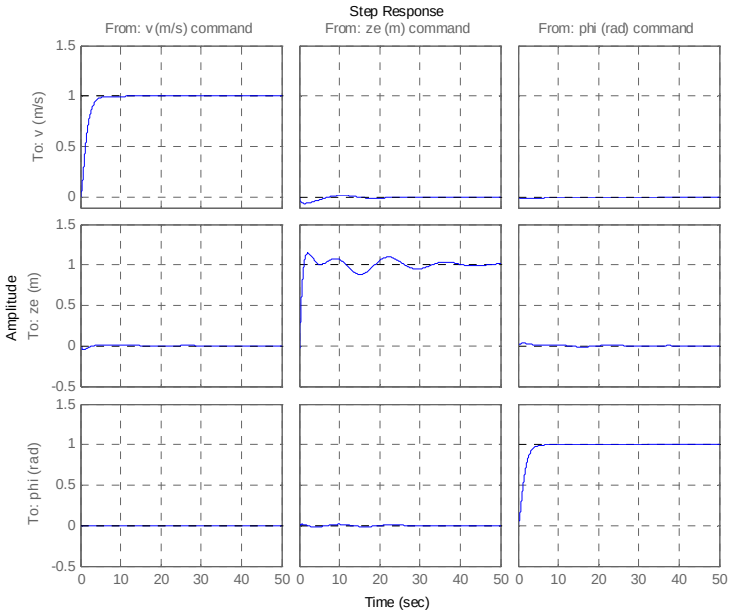

Figure 14. Aircraft states resulting from simulations of the linearized aircraft model in closed-loop under rudder jam around $v = 65\ m/s$ and $h = 1000\ m$.

The final step is to test the autopilot designed in closed loop with the nonlinear aircraft model from Airlib and perform SIMULINK simulations. Again we first wrap the controller designed above with an outer controller (see Figure 6) to an appropriate roll angle command ϕ_{cmd} from a desired direction of motion χ_{cmd} using a simple proportional integral derivative (PID) controller with filtered derivative of the form (21) where the coefficients were selected empirically as $K_p = 4$, $K_i = 1$, $K_d = 1$ and $N = 100$. This outer controller, the inner controller designed above and the nonlinear Cessna 172 model from Airlib were connected as illustrated in Figure 6 and the resulting system was simulated in SIMULINK. The scenario studied in this simulation is as follows: The aircraft is at location $p_{current} = (0,0)$, with heading $\chi_{current} = 0°$, speed $v = 65\ m/s$ and altitude $z_e = 1000\ m$ where it experiences a servoactuator jam that leaves the rudder stuck at 10°. The aircraft must therefore make an emergency landing. The closest runway starts at location $p_{runway} = (32990\ m, 12500\ m)$ and the approach direction for the runway is $-30°$. In the first part of the scenario it is desirable to navigate the aircraft to a point that is at least $15000\ m$ of the runway, while approaching the runway in a straight manner with a direction of travel $\chi_{target} = -30° = -0.5236\ rad$. In the second part of the scenario we would like to execute landing procedures from this point, which includes reducing the speed of the aircraft while at the same time decreasing its altitude. For the first part of the scenario we would like the aircraft to maintain its speed and altitude such that $v_{cmd} = 65\ m/s$ and $z_{e,cmd} = 1000\ m$. The point p_{target} to which the aircraft must be navigated can be obtained as follows:

$$\begin{aligned} p_{target} + \mathbf{1}_{-30°} \times 15000 &= p_{runway} \\ p_{target} &= (32990, 12500) - (\cos(-30°), \sin(-30°)) \times 15000 \\ p_{target} &= (20000, 20000) \end{aligned} \tag{26}$$

where $\mathbf{1}_{-30°}$ denotes the unit vector having a direction of $-30°$. The aircraft maneuver to reach this point can be achieved in three steps: a turn towards the target direction, straight travel, and a second turn to align the travel direction with the runway. Since the aircraft is disabled and flying at an usual attitude, the turns should not be too steep; hence we shall execute shallow turns with a wide turn radius of $r = 5000\ m$ for safety purposes. We therefore construct two circles with radius $r = 5000\ m$, one passing through $p_{current}$ with a tangent in the direction $\chi_{current}$ and another passing through p_{target} with a tangent in the direction χ_{target}. We then join these circles with a common tangent through which the aircraft will travel between turns. The direction of this common tangent and

the points of tangency to the circles can be computed by geometric calculations (or graphically) to be $\chi_{tangent} = 60.56^{\circ}$, $p_{t1} = (4355, 2543)$ and $p_{t2} = (13145,18126)$. Hence the χ_{cmd} to be applied to the autopilot first goes from $\chi_{current} = 0$ to $\chi_{tangent} = 60.56^{\circ} = 1.057\, rad$, then stays constant at $\chi_{tangent} = 60.56^{\circ} = 1.057\, rad$, and then goes from $\chi_{tangent} = 60.56^{\circ} = 1.057\, rad$ to $\chi_{target} = -30^{\circ} = -0.5236\, rad$. To compute the amounts of time t_1, t_2 and t_3 spent in the first, second and third segments mentioned above

$$\begin{aligned} \text{Duration of first turn: } t_1 &= \frac{|\chi_{tangent} - \chi_{start}|.r}{v} = \frac{|1.057 - 0|.5000}{65} \approx 81\ s \\ \text{Duration of straight travel: } t_2 &= \frac{\|p_{t2} - p_{t1}\|}{v} = \frac{\sqrt{(18126-2543)^2+(13145-4355)^2}}{65} \approx 275\ s \\ \text{Duration of second turn: } t_3 &= \frac{|\chi_{target} - \chi_{tangent}|.r}{v} = \frac{|-0.5236 - 1.057|.5000}{65} \approx 121\ s \end{aligned} \tag{27}$$

From this point on the landing procedure is executed, which includes reducing the speed of the aircraft uniformly from $v = 65\ m/s$ to its landing speed of $v_L = 34\ m/s$, while at the same time decreasing the altitude from $z_e = 1000\ m$ to $z_L = 0\ m$. The amount of time t_L spent during the landing phase can be computed as

$$\text{Duration of landing: } t_L = \frac{\|p_{target} - p_{runway}\|}{0.5(v+v_L)} = \frac{\sqrt{(20000-32990)^2+(20000-12500)^2}}{0.5(65+34)} \approx 303\ s \tag{28}$$

The altitude is decreased linearly until about the last minute before touchdown, and from this point on the descent rate is slowly decreased to achieve a flare. The results of the SIMULINK simulation of the controlled nonlinear aircraft of the scenario described above is presented in Figures 15 and 16. In Figure 15, the commands applied to the autopilot are shown in green dotted lines, and the aircraft's response is in blue solid lines, from where it can be seen that desired references are followed quite closely. The trajectory of the aircraft in is shown in Figure 16 where only the x_e and y_e components are present since the altitude is constant at $z_e = 1000\ m$ during the entire scenario. It can be seen that the aircraft follows the reference trajectory closely and reaches a point where it is about the desired distance (15000 m) from the target runway that is in alignment with the orientation of the runway. From this point on the landing procedure is executed, and the aircraft lands safely to the target runway.

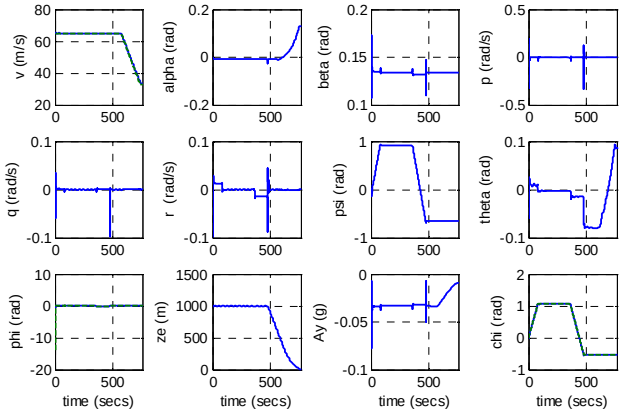

Figure 15. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under rudder jam.

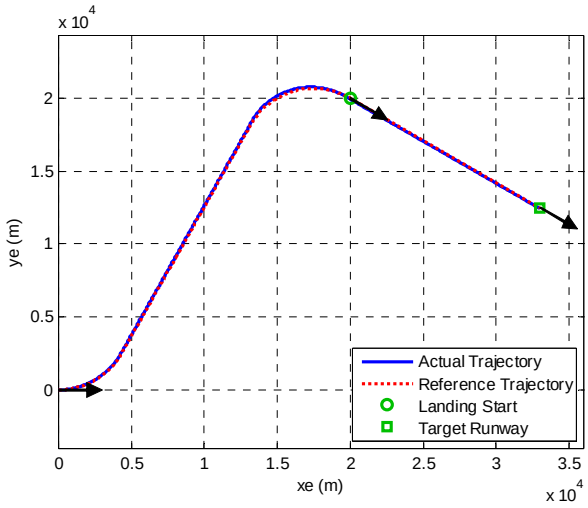

Figure 16. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under rudder jam.

C. Aileron Servoactuator Jam

In this section we consider the case when the aileron servoactuator is jammed, and for the sake of the example, we assume that the aileron is stuck at an angle of five degrees (i.e. $\delta_r = 5^\circ = 0.0873\ rad$) while the aircraft is flying at a cruise speed of $65\ m/s$ and an altitude of $1000\ m$. The uncontrolled response of the aircraft obtained from SIMULINK simulations using the nonlinear Cessna 172 model from Airlib is shown in Figures 17 and 18.

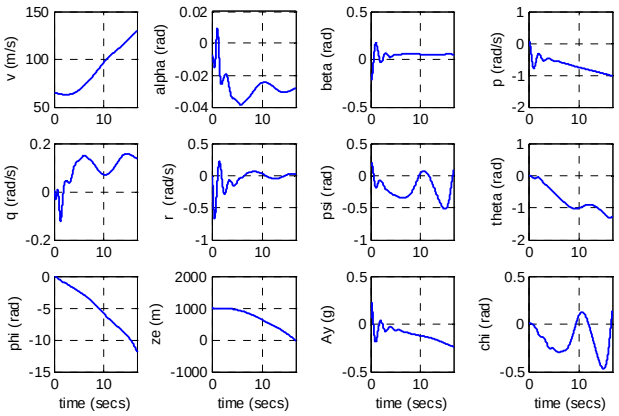

Figure 17. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model under aileron jam around $v = 65\ m/s$ and $h = 1000\ m$.craft states resulting from SIMULINK simulations of the nonlinear aircraft model for normal flight at $v = 65\ m/s$ and $h = 1000\ m$.

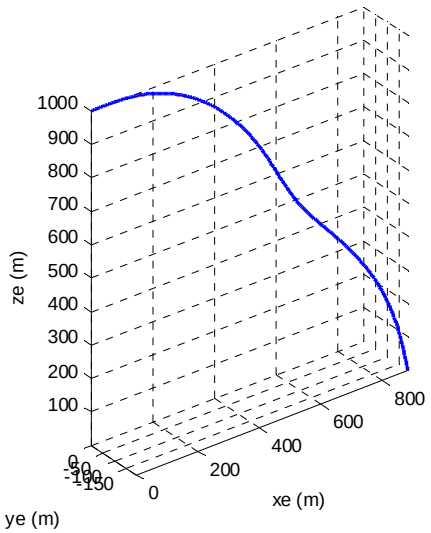

Figure 18. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model under aileron jam around $v = 65\ m/s$ and $h = 1000\ m$.

Similar to the rudder servoactuator jam case, the aircraft starts rolling around its axis, loses altitude and hits the ground at a high speed in less than 20 seconds. For control design the aircraft is first trimmed using the optimization routines under the Control and Estimation Tools Manager of MATLAB (see Figure 3) and using the functions of the Airlib library to compute trim points satisfying the operating specifications, which are as follows:

$$\mathrm{v} = 65\ \mathrm{m/s},\ \ \mathrm{z_e} = 1000\ \mathrm{m}, \delta_\mathrm{a} = 0.0873\ \mathrm{rad},\ \ \frac{\partial \alpha}{\partial \mathrm{t}} = \frac{\partial \beta}{\partial \mathrm{t}} = \frac{\partial \mathrm{p}}{\partial \mathrm{t}} = \frac{\partial \mathrm{q}}{\partial \mathrm{t}} = \frac{\partial \mathrm{r}}{\partial \mathrm{t}} = \frac{\partial \psi}{\partial \mathrm{t}} = \frac{\partial \theta}{\partial \mathrm{t}} = \frac{\partial \phi}{\partial \mathrm{t}} = \frac{\partial \mathrm{z_e}}{\partial \mathrm{t}} = 0 \qquad (29)$$

The trim point resulting for this specification is as follows:

$$\begin{aligned} x_0 &= [v, \alpha, \beta, p, q, r, \psi, \theta, \phi, x_e, y_e, z_e] \\ &= [65\ , -0.0073725, -0.22252, 0, 0, 0, 0.22252, 0.0049416, -0.0544, 0, 0, 1000] \\ u_0 &= [F_x, \delta_e, \delta_a, \delta_r] = [1251.3, -0.0065926, 0.087266, -0.29055] \end{aligned} \qquad (30)$$

where x_o is the aircraft state at the trim condition and u_0 is the input vector containing the throttle and surface deflections to the applied. Comparing with the trim point above with that of the normal flight given in (14), one sees again that it is necessary to increase the throttle (i.e. increase F_x) and also apply rudder input (δ_r) in the opposite direction to compensate for the aileron lock. Also similar to the rudder jam case, altitude the aircraft needs to fly at an unusual attitude to maintain the desired airspeed altitude (note the non-zero values for the angles β and ϕ). Since the aileron is unusable, we have only three control inputs (F_x, δ_e and δ_r) left and as before we set our three outputs to be controlled as $y = [v, z_e, \chi]$. As usual, we will first design an inner controller to set the roll angle (ϕ) to a desired value, and then wrap an outer controller to send the command ϕ_{cmd} to this inner controller based on the direction of travel command χ_{cmd}. The closed loop system structure is identical to that in Figure 6, except that the lateral acceleration A_y and its command $A_{y,cmd}$ are no longer part of the picture. As before, for inner controller design we first linearize the nonlinear aircraft model about the trim conditions (x_0, u_0), which yields a linear state-space system G of the form where the output vector y is $y = [v, z_e, \chi]$. The open loop step response for the linearized system is shown in Figure 19.

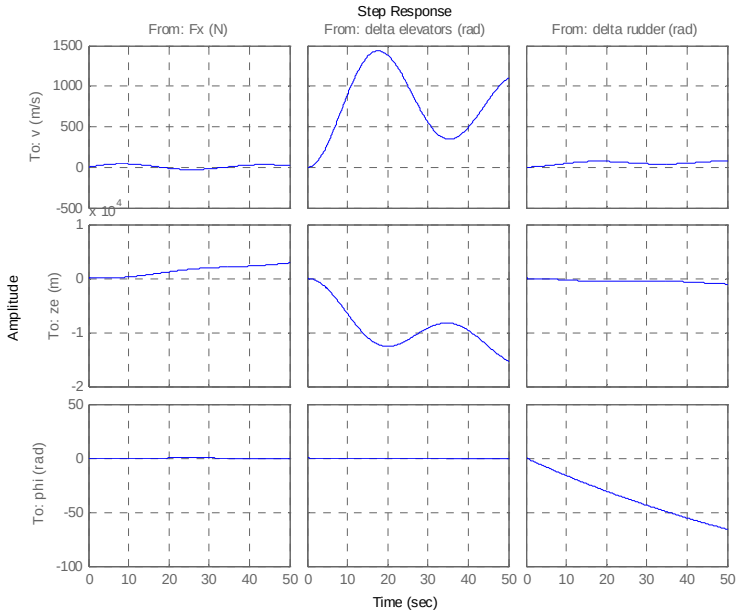

Figure 19. Aircraft states resulting from simulations of the linearized aircraft model in open-loop under aileron jam around $v = 65\ m/s$ and $h = 1000\ m$.

To design the inner controller we use the loop-shaping approach described in the previous section with a desired loop-shape

$$G_d(s) = \text{diag}\left(\frac{1}{s}, \frac{1}{s}, \frac{1}{s}\right) \tag{31}$$

The closed-loop transfer function for the inner loop resulting from this target loop-shape will approximately be

$$T(s) = G_d(s)[I + G_d(s)]^{-1} \approx \text{diag}\left(\frac{1}{s+1}, \frac{1}{s+1}, \frac{1}{s+1}, \frac{1}{s+1}\right) \tag{32}$$

Note that the settling time is $t_s = 5\tau = 5\frac{1}{1} = 5$ seconds, which is twice that of the normal flight scenario. The closed loop step response of the system is shown in Figure 20. Note that the diagonal plots converge to one, which implies that the desired tracking is achieved, and the off-diagonal plots are close to zero, which implies that the coupling effects between channels is negligible in closed-loop.

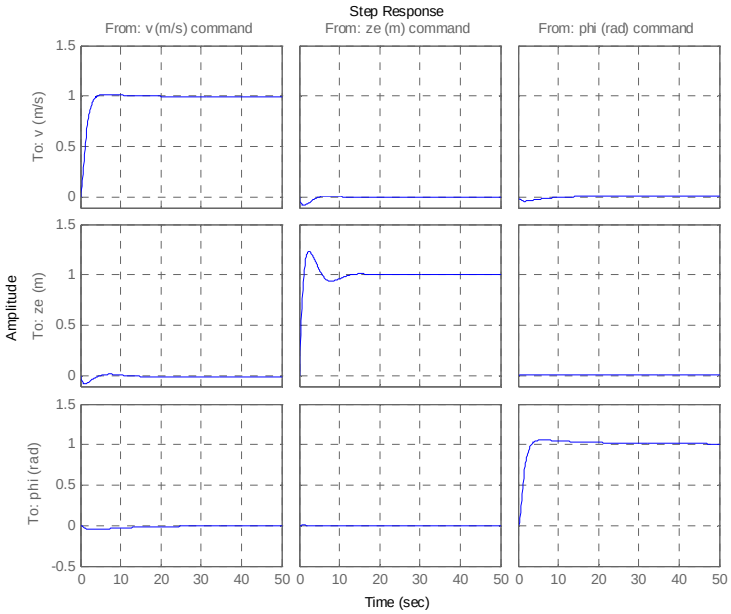

Figure 20. Aircraft states resulting from simulations of the linearized aircraft model in closed-loop under aileron jam around $v = 65\ m/s$ and $h = 1000\ m$.

The final step is to test the autopilot designed in closed loop with the nonlinear aircraft model from Airlib and perform SIMULINK simulations. Again we first wrap the controller designed above with an outer controller (see Figure 6) to an appropriate roll angle command ϕ_{cmd} from a desired direction of motion χ_{cmd} using a simple proportional integral derivative (PID) controller with filtered derivative of the form (21) where the coefficients were selected empirically as $K_p = 2.5$, $K_i = 0.12$, $K_d = 2$ and $N = 100$. This outer controller, the inner controller designed above and the nonlinear Cessna 172 model from Airlib were connected as illustrated in Figure 6 and the resulting system was simulated in SIMULINK. The scenario studied is the same as that in rudder jam case. The results of the SIMULINK simulation of the controlled nonlinear aircraft of the scenario described above is presented in Figures 21 and 22. In Figure 21, the commands applied to the autopilot are shown in green dotted lines, and the aircraft's response is in blue solid lines, from where it can be seen that desired references are followed rather closely. The trajectory of the aircraft in is shown in Figure 22; only the x_e and y_e components are shown since the altitude is constant at $z_e = 1000\ m$ during the entire scenario. It can be seen that the aircraft follows the reference trajectory closely and reaches a point where it is about the desired distance (15000 m) from the target runway and is in alignment with the orientation of the runway. From this point on the landing procedure is executed, and the aircraft lands safely to the target runway.

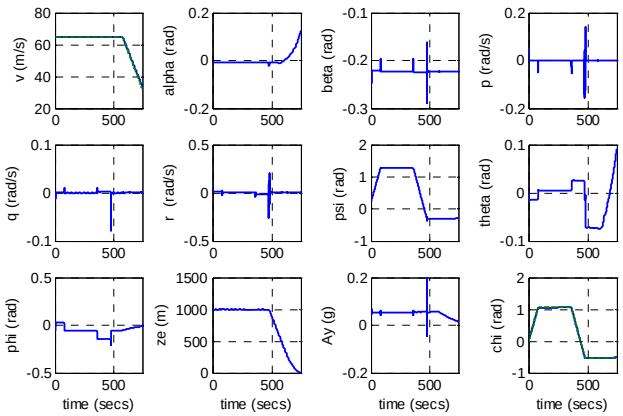

Figure 21. Aircraft states resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under aileron jam.

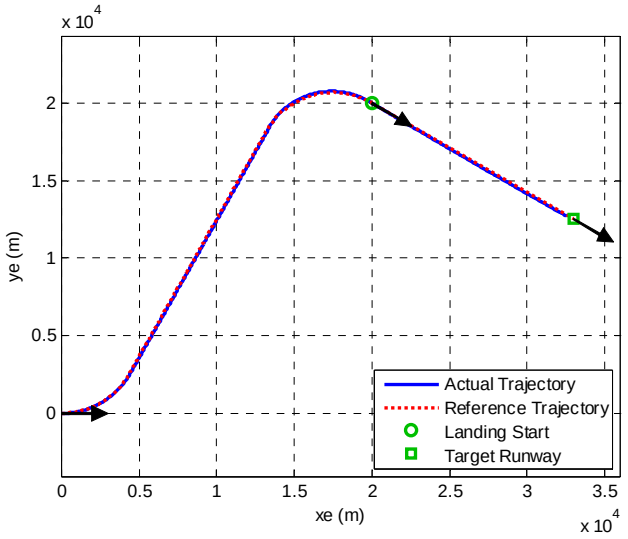

Figure 22. Aircraft trajectory resulting from SIMULINK simulations of the nonlinear aircraft model in closed-loop under aileron jam.

IV. Conclusion

Autopilot design for a disabled general aviation aircraft as a result of a rudder or aileron actuator jam is studied. Flight control laws that are required to automatically stabilize and then navigate the aircraft to a nearby airport are developed. Firstly, an accurate nonlinear aircraft model of the general aviation aircraft is introduced and its effects are simulated by using the MATLAB/SIMULINK incorporating the Airlib toolbox. The autopilot design was carried out by using loop-shaping techniques where decoupling of the longitudinal and lateral dynamics are explicitly addressed. Subsequently, an abnormal flight situation is shown where a sudden rudder servoactuator jam, if uncontrolled, leads to the eventual loss of the aircraft. However, it is shown that the newly designed autopilot is able to perform the necessary attitude change to first recover and then navigate the aircraft towards a nearby runway for an emergency landing. After a rudder malfunction, a similar study is carried out for the case of an aileron actuator jam where the autopilot can again automatically recover and navigate the aircraft. Future research directions include

extending the autopilot functions to demonstrate the full landing capability of a disabled aircraft, and using alternative approaches for controller design such as the nonlinear dynamical inversion.

Acknowledgements

The authors would like to thank Dr. Marc Rauw for *The Flight Dynamics and Control* (*FDC*) toolbox and Dr. Giampiero Campa for the *Airlib* toolbox, both of which have been used extensively in this work.

References

[1]National Transportation Safety Board, Aircraft Accident Report, PB90-910406, NTSB/ARR-90/06, United Airlines Flight 232, McDonnell Douglas DC-10, Sioux Gateway Airport, Sioux City, Iowa, July 1989.

[2]The Joint Unmanned Combat Aircraft Systems (J-UCAS) program sponsored by DARPA. http://www.rockwellcollins.com/athena/demos/damage-tolerance/

[3]Brady C., "The Rudder Story", The Boeing 737 Technical Guide, 2007. http://www.b737.org.uk/rudder.htm

[4]Burcham, Frank W., Jr., Trindel A. Maine, C. Gordon Fullerton, and Lannie Dean Webb, "Development and Flight Evaluation of an Emergency Digital Flight Control System Using Only Engine Thrust on an F-15 Airplane", NASA TP-3627, Sept. 1996.

[5]Gundy-Burlet, K., Krishnakumar, K., Limes, G., Bryant, D., "Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft". JACIC 2004, 1542-9423 Vol.1 no.12 (526-542).

[6]Kaneshige, John, John Bull, and Joseph J. Totah, "Generic Neural Flight Control and Autopilot System", AIAA 2000-4281, August 2000.

[7]Rau, M., "FDC 1.2 – A Simulink Toolbox for Flight Dynamics and Control Analysis", 2nd Edition, May 10, 2001, http://www.dutchroll.com.

[8]Rysdyk, Rolf T., and Anthony J. Calise, "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation", AIAA 98-4483, August 1998.

[9]Stevens, B.L. and Lewis, F.L., "Aircraft Control and Simulation," Wiley, 2e, 2003.

[10]Tjee, R.T.H., Mulder, J.A., "Stability and Control Derivatives of the De Havilland DHC-2 Beaver Aircraft," Report LR-556, Delft University of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands, 1988

[11]Gage, S., "Fly a Plane", September 22, 2004, http://www.mathworks.com/matlabcentral/fileexchange/3114, MATLAB Central.

[12] Le, V.X., and Safonov, M.G., "Rational matrix GCD's and the design of squaring-down compensators--a state space theory". IEEE Trans. Autom.Control, AC-36(3):384-392, March 1992.

[13]Gundy-Glover, K., and McFarlane D., "Robust stabilization of normalized coprime factor plant descriptions with H-bounded uncertainty". IEEE Trans. Autom. Control, AC-34(8):821-830, August 1992.

[14]Chiang, R.Y., and Safonov, M.G., "H-infinity synthesis using a bilinear pole-shifting transform". AIAA J. Guidance, Control and Dynamics, 15(5):1111-1115, September-October 1992.

[15]Campa, G., "Airlib", February 13, 2003, http://www.mathworks.com/matlabcentral/fileexchange/3019-airlib, MATLAB Central.