HOST INSTITUTION,

UNIVERSITY OF CRAIOVA

Project no. 89/1.10.2015

Program:	**Human Resources**
Project type:	**Young Teams**
Project Cods:	**PN-II-RU-TE-2014-4-0849**

Project title:

MODERN ARCHITECTURES FOR THE CONTROL OF AIRCRAFT LANDING

2015-2017

- SYSNTHESYS OF THE WORK -

SYSNTHESYS OF THE WORK,

containing the activities and the obtained results in comparison with the objectives of the project

for

- Stage I (2015) -

Stage I (2015) – ***Documentation regarding aircraft landing and the design of the sensors and multiple observers' analytical models***, took 3 months (October – December 2015); in this period, one has completely performed all the **5 activities**

I.1. Documentation regarding the aircraft dynamics, the errors of the sensors and the atmospheric disturbances during landing.

I.2. Documentation regarding the modern methods, optimal and adaptive, for the control of aircarft during landing.

I.3. The design of the analitical models of the sensors.

I.4. Design of the multiple observers for the longitudinal and lateral motions of aircraft during landing.

I.5. Design of the web page associated to the project – dissemination of the results.

All the **6 specific objective** have been acomplished:

OS1. To know aircraft dynamics, the error sensors, and all the atmospheric disturbances during landing.

OS2. To know the existing optimal and adaptive approaches for aircraft control during landing.

OS3. To design analytical models of the sensors taking into account both the deterministic and the stochastic errors.

OS4. To design a new multiple-observer for aircraft longitudinal motion during landing.

OS6. To design a new multiple-observer for aircraft motion during landing.

OS21. To disseminate the results in the scientific, academic and socio-economic environment.

Beside the acomplish of the scientific activities, the members of **the team achieved administrative and management activities (elaboration of scientific rapports, tasks distribution, tracking deadlines etc.), which competed at the completion of this stage in good condition.** Also, there were **regular meetings** between team members of the team, especially given that the two PhD students involved in the project are PhD students of "Politehnica" University of Bucharest. Also, the research team was met in full in three meetings to analyze activities and results so far achieved and to determine future actions of each member program. Funds allocated at this stage both for mobility, logistic and staff costs were managed successfully so that all objectives of the stage have been achieved. Below, it is presented the synthetic work done and results achieved in each of the eight activities intended to be conducted at this stage.

Activity I.1. Documentation regarding the aircraft dynamics, the errors of the sensors and the atmospheric disturbances during landing

There are three phases in a typical landing procedure: initial approach, glide slope, and flare [1]. During initial approach, the pilot descends from the cruise altitude to an altitude of approximately 420 m above the ground for heavy aircraft or less than 420 m for light aircraft. The pilot then positions the airplane so that the airplane is on a heading towards the runway centerline. As the airplane descends along the glide slope path, its pitch, attitude, and speed must be controlled; the aircraft maintains a constant speed along the flight path. The descent rate, for a Boeing 747, must be about 3 m/s and the pitch angle is between -5 to 5 degrees. As the airplane descends to 7-30 m above the ground (the maximum value is for Boeing 747), the slope angle control system is disengaged and a flare maneuver is executed. The vertical descent rate is slightly decreased so that the landing gear may be able to dissipate the energy of the impact at landing. The pitch angle of the airplane is then adjusted, between 0 to 5 degrees for most aircraft, which allows a soft touchdown on the runway surface [2].

For the ***glide slope phase*** ($H \geq H_0, H_0$ – the altitude at which the glide slope phase ends and the second landing phase begins), the commanded (calculated) altitude H_c and the real altitude H are, respectively: $\dot{H}_c = V_0 \sin\gamma_c \cong V_0\gamma_c$, $\dot{H} = V_0 \sin\gamma \cong V_0\gamma = V_0(\theta - \alpha)$, where $\gamma = \theta - \alpha$ is the real slope angle of the aircraft trajectory during landing, $\gamma_c = \theta_c - \alpha$ is the commanded value of this angle during the first stage of landing $(\gamma_c = -2.5\deg)$, V_0 – the nominal flight speed, α – aircraft attack angle, θ – aircraft pitch angle; because the slope angle, expressed in radians, has small values, the approximations $\sin\gamma \cong \gamma$ and $\sin\gamma_c \cong \gamma_c$ has been used. The dynamics associated to the ***flare phase*** $(H < H_0)$ has been presented in detail in [3]; the form of the calculated altitude H_c is $H_c = H_0 \exp(-t/\tau)$, with τ – the time constant that defines the exponential curve (flare landing phase). Let us consider H_{ref} – the altitude of the flare's final point (chosen below ground level such that the exponential trajectory associated to the flare intersects the runway in the desired point) [3]; therefore, the real altitude (H) is calculated with respect to H_{ref}, the relation between H and H_c being $H_c = H - H_{ref}$; by derivation of the above equation of H_c, we obtain: $\dot{H}_c = -\frac{1}{\tau}\underbrace{H_0 \exp(-t/\tau)}_{H_c} = -\frac{1}{\tau}H_c$. Taking into account that $H_c = H - H_{ref}$, the solution of the previous differential equation is: $H_c = H_0 - \frac{1}{\tau}\int_0^{t_c}(H - H_{ref})\mathrm{d}t$, where t_c is the flare's duration. Now, by using the previous equations, one can obtain the calculated value (reference value) of the aircraft pitch angle (θ_c) for the two stages of landing; it has the expressions:

$$\theta_c \cong \alpha + \frac{\dot{H}_c}{V_0} \cong \begin{cases} \alpha + \gamma_c, H \geq H_0, \\ \alpha + \frac{1}{V_0\tau}(H_{ref} - H), H < H_0. \end{cases} \tag{1.1}$$

The dynamics used in this paper belongs to a Boeing 747; the linear model of the aircraft motion, in longitudinal plane, is described by the state equation [3], [4]: $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + G\boldsymbol{u}_w$, with $\boldsymbol{x} \in R^{4\times1}$ – the state vector, $\boldsymbol{x} = [u \quad w \quad q \quad \theta]^T, \boldsymbol{u} \in R^{2\times1}$ – the command vector, $\boldsymbol{u} = [\delta_e \quad \delta_T]^T$, while $\boldsymbol{u}_w$ is the vector of disturbances which is estimated by the equipment from aircraft's navigation system; u is the aircraft longitudinal velocity, w - aircraft vertical velocity, q – aircraft pitch angular rate, θ - aircraft pitch angle, while δ_e and δ_T are the elevator deflection and the engine command, respectively. Taking into account that w is much smaller than u, the aircraft velocity in longitudinal plane can be approximated as follows: $V = \sqrt{u^2 + w^2} \cong u$; thus, the nominal value of V is considered to be $V_0 \cong u(0) = u_0$. The matrices $A \in R^{4\times4}, B \in R^{4\times2}$ are, respectively [5]: $A = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ 0 & 0 \end{bmatrix}$. If one extends the state vector $\boldsymbol{x}$ with two new states δ_e and δ_T (the outputs of the actuators), satisfying the equations: $\dot{\delta}_e = -\frac{1}{T_e}\delta_e + \frac{1}{T_e}\delta_{ec}, \dot{\delta}_T = -\frac{1}{T_T}\delta_T + \frac{1}{T_T}\delta_{Tc}$, then, the new state of the system becomes $\boldsymbol{x} = \begin{bmatrix} \frac{u}{V_0} & \frac{w}{V_0} & q & \theta & \delta_e & \delta_T \end{bmatrix}^T$, while the new command vector is $\boldsymbol{u} = [\delta_{ec} \quad \delta_{Tc}]^T$; δ_{ec} and δ_{Tc} are the commands applied to elevator and to engines, respectively. Therefore, for $\boldsymbol{u}_w = 0$, the aircraft state equations are:

$$\begin{gathered} \frac{\dot{u}}{V_0} = a_{11}\frac{u}{V_0} + a_{12}\frac{w}{V_0} + \frac{a_{14}}{V_0}\theta + \frac{b_{11}}{V_0}\delta_e + \frac{b_{12}}{V_0}\delta_T, \frac{\dot{w}}{V_0} = a_{21}\frac{u}{V_0} + a_{22}\frac{w}{V_0} + \frac{a_{23}}{V_0}q + \frac{a_{24}}{V_0}\theta + \frac{b_{21}}{V_0}\delta_e + \frac{b_{22}}{V_0}\delta_T, \\ \dot{q} = V_0 a_{31}\frac{u}{V_0} + V_0 a_{32}\frac{w}{V_0} + a_{33}q + a_{34}\theta + b_{31}\delta_e + b_{32}\delta_T, \dot{\theta} = q, \dot{\delta}_e = -\frac{1}{T_e}\delta_e + \frac{1}{T_e}\delta_{ec}, \dot{\delta}_T = -\frac{1}{T_T}\delta_T + \frac{1}{T_T}\delta_{Tc}. \end{gathered} \tag{1.2}$$

Having in mind that $w = V_0 \sin\alpha$ or $w \cong V_0\alpha$ for small values of the attack angle $(\sin\alpha \cong \alpha)$, for the aircraft control during the glide slope phase, the following state $\boldsymbol{x} \in R^{8\times1}$ is chosen: $\boldsymbol{x} = \begin{bmatrix} \frac{u}{V_0} & \alpha & q & \theta & \frac{H}{V_0} & \frac{\dot{H}}{V_0} & \delta_e & \delta_T \end{bmatrix}^T$. Thus, to equations (1.2), one adds the differential equations associated to the variables H/V_0 and $\dot{H}/V_0$; these are:

$$\begin{aligned} &\frac{\dot{H}}{V_0} \cong \theta - \alpha, \\ &\frac{\ddot{H}}{V_0} = -a_{21}\frac{u}{V_0} - a_{22}\alpha + \left(1 - \frac{a_{23}}{V_0}\right)q - \frac{a_{24}}{V_0}\theta - \frac{b_{21}}{V_0}\delta_e - \frac{b_{22}}{V_0}\delta_T; \end{aligned} \tag{1.3}$$

the last one has been obtained by derivation with respect to time of the first equation (1.3), with $\dot{\alpha} = \dot{w}/V_0$ having the form (1.2).

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \frac{a_{14}}{V_0} & 0 & 0 & \frac{b_{11}}{V_0} & \frac{b_{12}}{V_0} \\ a_{21} & a_{22} & \frac{a_{23}}{V_0} & \frac{a_{24}}{V_0} & 0 & 0 & \frac{b_{21}}{V_0} & \frac{b_{22}}{V_0} \\ V_0 a_{31} & V_0 a_{32} & a_{33} & a_{34} & 0 & 0 & b_{31} & b_{32} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -a_{21} & -a_{22} & \left(1 - \frac{a_{23}}{V_0}\right) & -\frac{a_{24}}{V_0} & 0 & 0 & -\frac{b_{21}}{V_0} & -\frac{b_{22}}{V_0} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_e} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_T} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_e} & 0 \\ 0 & \frac{1}{T_T} \end{bmatrix}. \tag{1.4}$$

Denoting with $\boldsymbol{u_w} = \begin{bmatrix} u_g/V_0 & \alpha_g & q_g \end{bmatrix}^T$ – the vector of disturbances which are additionally introduced in the equations of the states $u/V_0, \alpha, q$, one yields the matrix G as follows:

$$G^T = \begin{bmatrix} a_{11} & a_{21} & V_0 a_{31} & 0 & 0 & -a_{21} & 0 & 0 \\ a_{12} & a_{22} & V_0 a_{32} & 0 & -1 & -a_{22} & 0 & 0 \\ 0 & a_{23}/V_0 & a_{33} & 1 & 0 & (1 - a_{23}/V_0) & 0 & 0 \end{bmatrix}. \tag{1.5}$$

The first component of the vector $\boldsymbol{u_w}$ refers to the longitudinal gust, while the second one (α_g) refers to the vertical gust taking into account that $\alpha \cong w/V_0$ and, accordingly, $\alpha_g \cong w_g/V_0$; w_g is the speed of the vertical wind (disturbance).

For the aircraft control during flare, using the notation $\widetilde{H}_c = \int_{t_o}^{t} H_c(\tau)\mathrm{d}\tau$ (t_0 – the time moment when the aircraft ends the glide slope phase and begins the flare phase), the following state $\boldsymbol{x} \in R^{8\times1}$ is chosen: $\boldsymbol{x} = \begin{bmatrix} \frac{u}{V_0} & \alpha & q & \theta & \frac{H}{V_0} & \frac{\widetilde{H}_c}{V_0} & \delta_e & \delta_T \end{bmatrix}^T$; matrix A is obtained with (1.4) modifying only the sixth line, while the matrix B is the same with the one in (1.4); thus, one gets:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \frac{a_{14}}{V_0} & 0 & 0 & \frac{b_{11}}{V_0} & \frac{b_{12}}{V_0} \\ a_{21} & a_{22} & \frac{a_{23}}{V_0} & \frac{a_{24}}{V_0} & 0 & 0 & \frac{b_{21}}{V_0} & \frac{b_{22}}{V_0} \\ V_0 a_{31} & V_0 a_{32} & a_{33} & a_{34} & 0 & 0 & b_{31} & b_{32} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_e} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_T} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_e} & 0 \\ 0 & \frac{1}{T_T} \end{bmatrix}. \tag{1.6}$$

Aircraft landing is simplified if the motion of the aircraft in lateral plane is made without errors (deviation of the aircraft from the runway direction is zero). This is why the system for the automatic control of the flight direction is very important. Before the start of the landing two main stages in longitudinal plane (the glide slope phase and the flare phase), the pilot must cancel the aircraft lateral deviation with respect to the runway.

The linearization of an aircraft nonlinear dynamics is generally based on the method of small disturbances with respect to an equilibrium trajectory. The general linear model of the aircraft motion (**A**), in lateral-directional plane, is described by the equations [6]:

$$\begin{aligned}
&\dot{\beta} = a_{11}\beta + a_{12}p + a_{13}r + a_{14}\varphi + b_{11}\delta_a + b_{12}\delta_r + \frac{a_{11}}{V_0}V_{vy},\ \dot{p} = a_{21}\beta + a_{22}p + a_{23}r + b_{21}\delta_a + b_{22}\delta_r + \frac{a_{21}}{V_0}V_{vy},\\
&\dot{r} = a_{31}\beta + a_{32}p + a_{33}r + b_{31}\delta_a + b_{32}\delta_r + \frac{a_{31}}{V_0}V_{vy},\ \dot{\varphi} = p,\ \dot{\psi} = r,\ \dot{Y} = -V_0\beta + V_0\psi + V_{vy},\\
&\dot{\delta}_a = -\frac{1}{T_a}\delta_a + \frac{1}{T_a}\delta_{a_c},\ \dot{\delta}_r = -\frac{1}{T_r}\delta_r + \frac{1}{T_r}\delta_{r_c},
\end{aligned} \tag{1.7}$$

where β is the aircraft sideslip angle, φ and ψ are the roll angle and the yaw angle, respectively, p – the aircraft roll angular rate, r – the aircraft yaw angular rate, Y – the deviation of the aircraft with respect to the runway direction, δ_a and δ_r – the deflection angles of the ailerons and rudder, respectively, δ_{a_c} and δ_{r_c} – the roll and yaw commands, V_{vy} – the component of the wind velocity along the lateral axis of the aircraft, V_0 – aircraft nominal velocity, T_a and T_r – the effectors' time delay constants of the ailerons and rudder, respectively. Aircraft's lateral motion may be written under the form: $\dot{x} = Ax + Bu + Gw$; here x is the state vector, u - the command vector, while the crosswind $w = V_{vy}$ is the system's disturbance which is estimated by the equipment from aircraft's navigation system; the forms of x and u are $x = [\beta \quad p \quad r \quad \varphi \quad \psi \quad Y \quad \delta_a \quad \delta_r]^T$ and $u = [\delta_{a_c} \quad \delta_{r_c}]^T$, respectively. The following matrices result:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & b_{11} & b_{12} \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & b_{21} & b_{22} \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 0 & b_{31} & b_{32} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -V_0 & 0 & 0 & 0 & V_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_r} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_a} & 0 \\ 0 & \frac{1}{T_r} \end{bmatrix}, G = \begin{bmatrix} \frac{a_{11}}{V_0} \\ \frac{a_{21}}{V_0} \\ \frac{a_{31}}{V_0} \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{1.8}$$

The output equation associated to the aircraft dynamics during landing can be put under the form: $y = C\mathbf{x} + D_{22}e$, where $D_{22} = \bar{k}I_7$ ($\bar{k}$ – positive constant) is the matrix of weights associated to the vector containing the errors of the sensors [2]; for the glide slope, the vectors e and y are: $e = [e_u \quad e_\alpha \quad e_q \quad e_\gamma \quad e_H \quad e_{\delta_e} \quad e_{\delta_T}]^T, y = [u \quad \alpha \quad q \quad \gamma \quad H \quad \delta_p \quad \delta_T]^T$, while the matrix C is obtained through identification. For the flare phase, the forms of the vectors e and y become: $e = [e_u \quad e_\alpha \quad e_q \quad e_\theta \quad e_H \quad e_{\delta_e} \quad e_{\delta_T}]^T$ and $y = [u \quad \alpha \quad q \quad \theta \quad H \quad \delta_p \quad \delta_T]^T$; D_{22} has the same expression as above.

It is well known that the sensors are used for the measurement of different parameters have errors; e.g. the most important errors of the sensors are: 1) the bias; 2) the scale factor; 3) calibration error; 4) the noise; 5) sensitivity to accelerations aplied along some arbitrary directions. The biases are the most severe errors for the control of aircraft during landing.

For the girometric sensors, the error's model is described by the equation: $\omega = (\omega_i + S \cdot a_r + \bar{B} + \nu)\left(1 + \frac{\Delta K}{K}\right)$, where ω is the angular velocity (the disturbed signal – output signal), ω_i – angular velocity (input signal), S – sensitivity to the acceleration

a_r applied along with an arbitrary direction, $\overline{B}-$ the bias, $K-$ the scale factor, $\Delta K-$ the calibration error of the scale factor, $v-$ the noise of the sensor.

The implementation of the above equation in Matalb/Simulink leads to the model in fig. 1.1. One considered that the bias is given by its maximum value $\overline{B}$, the calibration error of the scale factor is given by its maximum value ΔK as percent from K, while the noise is given by the maximum value of its density. By using the function “randn(1)” one generates the bias through an arbitrary value in the interval $(-\overline{B}, \overline{B})$, the sensitivity S in the interval $(0, S)$ and the calibration error of the scale factor in the interval $(-\Delta K, \Delta K)$. The noise is generated by using the Simulink block “Band-Limited White Noise” and the Matlab function “RandSeed” which generates an arbitrary value of its density in the interval $(80\% \cdot \nu_d, \nu_d)$. The inputs of the model are the angular velocity ω_i, applied along the sensor’s sensitivity axis, and the acceleration a_r, considered to be the resultant acceleration of the vehicle, while the output is the disturbed angular velocity ω.

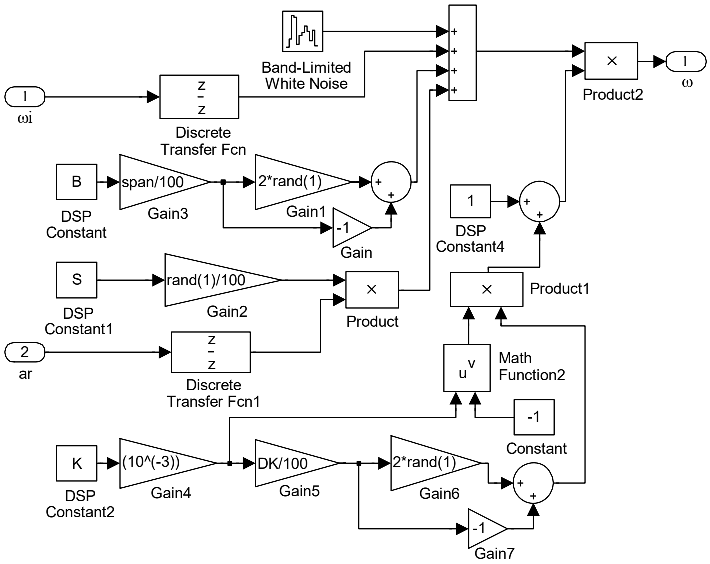


Fig. 1.1. Error model for girometers implementd in Matlab/Simulink

The models of atmospheric disturbances habe been introduced by Karman and Dryden and adopted by the International Civil Aviation Organization Standard Atmosphere. A determinist model which express the wind velocity is [3]:

$$V_{vz}=\begin{cases}0, \ H>470\,\mathrm{m}, \\ 2.86585 V_{v9.15}, \ 300\,\mathrm{m}<H\le 470\,\mathrm{m}, \\ V_{v9.15}\dfrac{H^{0.2545}-0.4097}{1.3470}, \ 0<H\le 300\,\mathrm{m},\end{cases} \tag{1.9}$$

where $V_{v9.15}$ is the wind velocity for the altitude H=9.15 m. A spectral model of the atmospheric disturbances is given by the following equations [4]: $V_{vx}=V_{vxc}+V_{vx1}, \dot{V}_{vx1}=0.2|V_{vxc}|\sqrt{2a_u}\,N_1-a_u V_{vx1}, V_{vxc}=\begin{cases}-U_0\left(1+\ln\dfrac{H}{510}/\ln 51\right), \ H\ge 510\,\mathrm{ft}, \\ 0, \ H<510\,\mathrm{ft},\end{cases}$ where U_0=20 ft/sec is the wind velocity for H=510 ft, N_1=100 is the intensity of the Gausian white noise $a_u=\begin{cases}V_0/\left(100\sqrt[3]{H}\right), \ H>230\,\mathrm{ft}, \\ V_0/600, \ H\le 230\,\mathrm{ft},\end{cases}$ with V_0 [ft/sec] – the nminal velocity of aircraft.

For the vertical components of the wind velocity V_{vz}, the calculation equations are similar; the exception is that in the equation of V_{vzc} the role of U_0 is played by W_0=40 ft/sec; the equation of V_{vz1} is second order, the place of a_u being taken by a_w=U_0/H,

$$\sigma_w=\sqrt{3}\begin{cases}0.2|V_{vxc}|, \ H>510\,\mathrm{ft}, \\ 0.2|V_{vxc}|(0.5+0.00098H), \ H\le 510\,\mathrm{ft}.\end{cases} \tag{1.10}$$

Another spectral model of the atmospheric disturbances is [5]

$$V_{vx} = V_{vxc} + N_1 \sqrt{\frac{1}{\Delta t}} \frac{\sigma_u \sqrt{2a_u}}{s + a_u}, V_{vz} = N_1 \sqrt{\frac{1}{\Delta t}} \frac{\sigma_w \sqrt{3a_w}\left(s + b_w\right)}{\left(s + a_w\right)^2}, \tag{1.11}$$

where Δt is the simulation time, $\sigma_u = 0.2\left|V_{vxc}\right|_0, a_w = V_0 / H, b_w = V_0 / \sqrt{3}H$, σ_w has the expression (1.10). The expression from parantheses express the transfer functions of the filters having white noise type input signals; these filters model (provides as outputs) the velocities induced by the atmospheric disturbances. In [6] the white noise is generated by transforming distribuited numbers through the transformations $V_v = \overline{V}_v + \sigma\sqrt{-2\ln x_1}\cos\left(2\pi x_2\right)$, where $\overline{V}_v$ is the wind medium velocity, σ – the standard deviation of the wind velocity, x_1 and x_2 – numbers between 0 and 1 randomly generated.

For Dryden turbulences, the transfer functions of the filters for the wind components (V_{vx}, V_{vy}, V_{vz}) are [6]:

$$H_u(s) = \sigma_u \sqrt{\frac{2L_u}{U_0}} \frac{1}{1 + \frac{L_u}{U_0}s}, H_v(s) = \sigma_v \sqrt{\frac{L_v}{\pi U_0}} \frac{1 + \frac{\sqrt{3}L_v}{U_0}s}{\left(1 + \frac{L_v}{U_0}s\right)^2}, H_w(s) = \sigma_w \sqrt{\frac{L_w}{\pi U_0}} \frac{1 + \frac{\sqrt{3}L_w}{U_0}s}{\left(1 + \frac{L_w}{U_0}s\right)^2}, \tag{1.12}$$

where $L_u = L_v = \dfrac{H}{\left(0.177 + 0.000823\,H\right)^{1.2}}, L_w = H; \sigma_u = \sigma_v = \dfrac{\sigma_w}{\left(0.177 + 0.000823\,H\right)^{0.4}}, \sigma_w = 0.1V_{20}, V_{20} = 20\,\text{ft/s}, [H] = \text{ft}$.

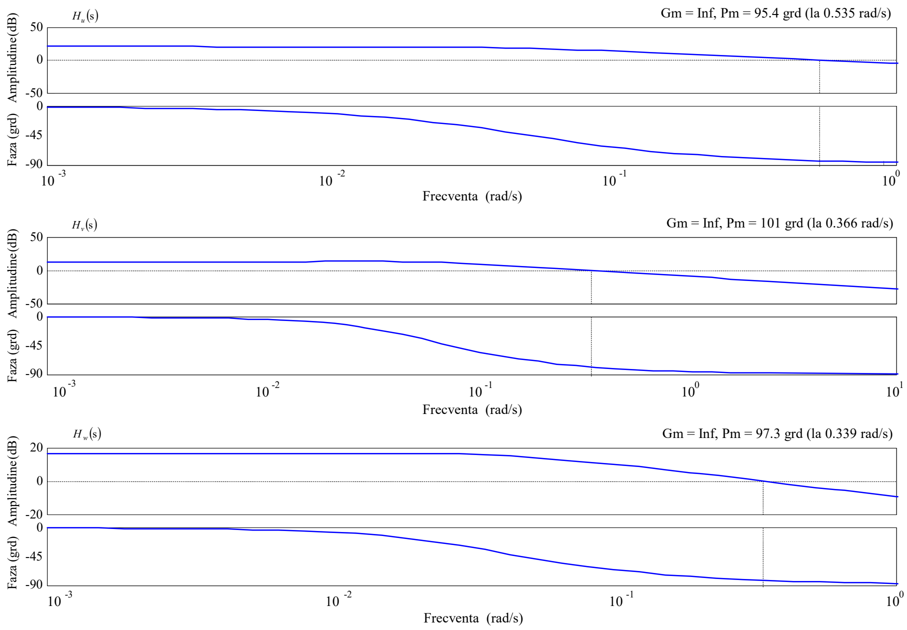


Fig. 1.2. Frequencey characteristics of the filters having the transfer functions $H_u(s), H_v(s), H_w(s)$

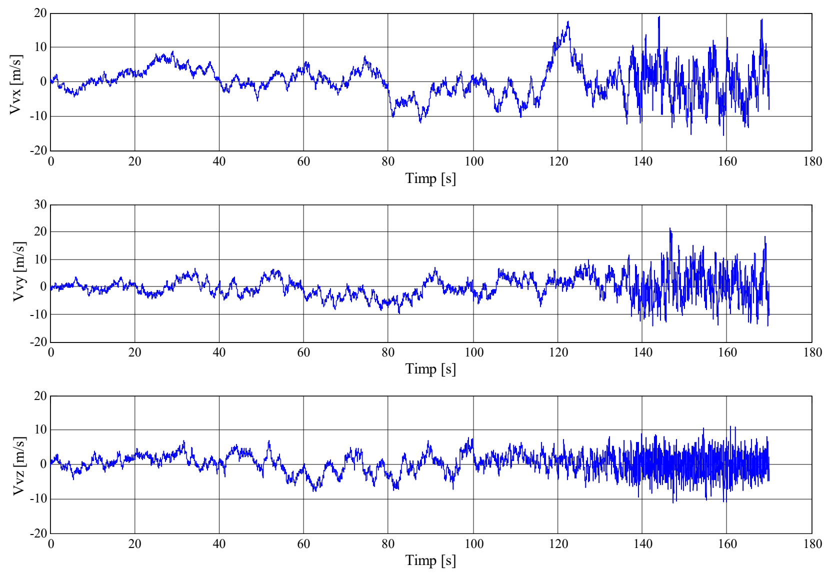


Fig. 1.3. Time characteristics V_{vx}, V_{vy}, V_{vz} for the Dryden spectral model f the atmospheric disturbances

In fig. 1.2 one presents the frequency characteristics for the filters having the transfer functions $H_u(s), H_v(s), H_w(s)$, at H=500 m şand U_0=50 ft/s. The cut frequencies for the three transfer functions are: 0.535 rad/s; 0.366 rad/s; 0.339 rad/s.

The wind shears in longitudinal plane are modeled by the following equations [7]:

$$V_{vx} = -V_{vx0}\sin\frac{2\pi t}{T_0}, V_{vh} = -V_{vh0}\left(1-\cos\frac{2\pi t}{T_0}\right), \tag{1.13}$$

where T_0 is the time spending in the wind shear, while V_{vx0} and V_{vh0} are the maximum values of the wind velocities along the longitudinal and lateral axes. The most used values for these are, respectively: T_0=60 s, V_{vx0}=12 ft/s, V_{vz0}=6 ft/s. In fig. 1.3 one represents the graphical characteristics V_{vx}, V_{vy}, V_{vz} for the Dryden spectral model associated to the atmospheric disturbances.

Activity I.2. Documentation regarding the modern methods, optimal and adaptive, for the control of aircarft during landing

In the ALSs' design process, different conventional control laws (PD, PI, PID) have been used, as well as different laws based on the state vector, dynamic inversion concept, with command filters, dynamic compensators, and state observers [1, 8-12]. The classical controllers cannot track satisfyingly the desired landing trajectory and velocity, especially when there is full of disturbances and uncertainties in flight, such as wind shears, atmospheric disturbances, and sensor errors; therefore, the optimal and adaptive ALSs took people's attention [13]. Recently, lots of researchers have applied the intelligent concepts for the aircraft automatic landing, by using the optimal synthesis H_2, H_∞, H_2/H_∞ [14, 15], the adaptive synthesis based on dynamic inversion and NNs [3, 7, 16], quantitative feedback theory [4], sliding mode control [17], linear quadratic optimal control (LQR/LQG), structured singular value μ-synthesis, or fuzzy techniques [18, 19]. Feed-forward neural networks with back propagation learning algorithm have also been used [10], the main drawback of such systems being that the NNs require a priori training on normal and faulty operating data. In [20] the feedback linearization method has been used for nonlinear control in the design of an automatic landing system; unfortunately, the paper presents limited insight into the performance of simulations of this controller and no tests are performed outside of these simulations. The algorithm (nonlinear control based on the dynamic inversion approach) obtained by Singh and Padhi [21], for the automatic landing of UAVs along with associated path planning, is not tested in the presence of sensor errors and external disturbances, this being a disadvantage of the algorithm. In the research area of optimal synthesis, Shue and Agarwal [13] have developed a mixed technique for the H_2/H_∞ control of landing, while Ochi and Kanai [22] have used the H_∞ control technique to design an approach for aircraft automatic landing. In these papers, the authors did not analyze the robustness of the designed controllers in the presence of sensor errors and external disturbances – issue which is considered in our project. The paper [7] presents an intelligent automatic landing system that uses a time delay neural network controller and a linearized inverse aircraft model to improve the performance of conventional automatic landing systems, a learning-through-time process being used in the controller training; the disadvantage of the designed ALS is that it is enabled only under limited conditions. If severe wind disturbances are encountered, the pilot must handle the aircraft due to the limits of the ALS. From the observers' point of view, it is known that the observers are useful in any control architecture to replace or augment some of the sensors [23, 24]; when a system has unknown inputs, the so-called "unknown input observers" are used. There are 3 important design methods to design such observers: geometrical methods [25], algebraic methods [26, 27], and methods that use the generalized inverse [28]. All these methods involves the usage of linear dynamics (by linearization) taking into account **only one operating point**.

The **technical/scientific barriers** which have prevented the widespread and use of optimal and adaptive ALSs, as well as **our solutions to these problems** are presented below.

No.	Technical/scientific barriers	Our solutions to these technical/scientific problems
1.	The linear observers' design methods use the linear dynamics – obtained, by linearization, taking into account only single operating point.	We will design and use in all the optimal and adaptive automatic landing systems the multiple-observers; these will be designed by means of multiple-models (based on many local linear models, any of them giving information about the system in a specific operating regime).
2.	None ALS designed till now considers in aircraft dynamics, in the same time, all the sensor errors together with wind shears and	This technical/scientific barrier will be lifted: the wind shears (having sinusoidal forms), the atmospheric turbulences (with Dryden spectral model), and the all the sensor errors will be modeled, considered in aircraft dynamics, software

	the atmospheric turbulences.	implemented as *Error blocks*, their influence being then quantified together with the ALSs' robustness increasing.
3.	In the present literature, only simplest models of the errors are considered; these are based on forced approximations, even empirical, being inadequate for a complete and complex study of the ALSs.	Beside the deterministic errors of the sensors (fixed values), we will take into account the variations of these errors with respect to environment conditions (e.g.: variation of the temperature). We will design complex analytical models of the sensors (software implemented) which will take into account, beside the deterministic errors, the stochastic ones.
4.	None ALS uses an algorithm to modify on-line the weights of the H_2 and H_∞ control techniques in the H_2/H_∞ control low; this can lead to the decrease of ALS's accuracy.	An on-line algorithm will be actualized and used; a quantifiable constant (k) will be modified on-line such that aircraft accurately tracks the desired trajectory during landing; if k=1, the control law is designed only using the H_∞ technique; if k=0, the control law is designed only by using the H_2 technique; if k is between 0 and 1, the two techniques will have different weights in the final control law.
5.	For the linearized motion of aircraft, separate ALSs are designed for the longitudinal and lateral planes.	After the design of the ALSs – longitudinal plane (control of glide slope and flare) and ALSs – lateral plane (systems for the automatic control of the flight direction), we will concatenate these two parts of the landing auto-pilot.
6.	In the adaptive automatic landing systems, the neural networks may have adapting difficulties.	PCH blocks will be designed and introduced in our new architectures; these will limit the pseudo-control (modify the signals of the NNs) and will "move back" the reference model (introducing a correction of the reference model's response with respect to the execution element's position). Also, we will design a new and modern way to on-line estimate the signal which must be provided by the PCH blocks using the fuzzy-logic.
7.	In the automatic landing systems, the controllers are sensitive to actuators' nonlinearities.	
8.	Some of the disturbances still influence significantly the landing trajectory.	Compensation algorithms (based on fuzzy logic) will be designed and additional compensation terms will be added in the controllers.

Activity I.3. The design of the analitical models of the sensors

Considering the diversity of size and scope of application for flight equipment, MAVs, drone and UAVs, to military and civilian aircraft, most inertial sensor manufacturing technologies can successfully compete in occupation a place in the system of detection of an automatic landing control system for aircraft. This can be seen from a simple overview of inertial sensor manufacturing technologies that can serve the various precision classes in air navigation: 1) The first class of precision is still under the dominance of laser and fiber optic of classical (mechanical) accelerometers, with the tendency for optical fiber and atomic interferometry to remove the laser and atomic interferometric accelerometry to enter easily in this area; 2) the second class of precision is reserved for optically integrated optical, interferometer and fiber optic and atomic interferometry technologies and classical (mechanical) accelerometer technologies with atomic interferometry and MEMS; 3) the third class of precision (equivalent to standard navigation applications) will be fully dominated by MEMS and optical integrated optical miniaturizers, MOEMS-oriented technologies; 4) the fourth precision class (military tactical applications) is expected to be fully dominated by MEMS and MOEMS accelerometer and gyrometers; 5) The fifth class of precision, related to commercial applications, is already recording a division of rights between MEMS, MOEMS, NEMS and NOEMS technologies.

The main sensors on an aircraft are: three accelerometers (for the measurement of the accelerations a_x, a_y, a_z, which, through integration, provide the velocities V_x, V_y, V_z) and three girometers (for the measurement of the angular velocities $\omega_x, \omega_y, \omega_z$) connected in an inertial navigation system (INS), a GPS sensor (for the measurement of the aircraft position), static and dynamic pressure sensors, a radio-altimeter or other device for the measurement of the altitude etc. The roll, pitch and yaw angles (φ, θ, ψ) can be obtained through the integration of the angular velocities $\dot{\varphi}, \dot{\theta}, \dot{\psi}$, which are the solutions of the system:

$$\dot{\varphi} = \omega_x + \omega_y \sin\varphi \operatorname{tg}\theta + \omega_z \cos\varphi \operatorname{tg}\theta, \dot{\theta} = \omega_y \cos\varphi - \omega_z \sin\varphi, \dot{\psi} = \omega_y \sin\varphi \sec\theta + \omega_z \cos\varphi \sec\theta. \quad (1.14)$$

The present activity aimed at developing analytical error models for the gyrometer and accelerometer sensors included in the hardware device that provides real-time data for landing control systems to aircraft. The analysis of the specialized literature and the experience of the research team acquired in previous projects related to the numerical and experimental development of stray-down inertial navigators led to the need to model both categories of errors affecting these sensors: deterministic errors and stochastic errors. The developed models allow for the reproduction of the accelerometer and gyrometer sensor errors in accordance with the values presented in the vast majority of the technical datasheets produced by the manufacturers but also by the standards used for carrying out their test and calibration procedures. New analytical models have been developed because standardized analytical models (especially IEEEs) are extremely cumbersome, with limited software implementation capabilities that offer good reproducibility of the sensor's real characteristics, and what is most important , they are characterized by a low degree of applicability from one category of sensors to another, such as, for example, the modeling of MEMS gyrometers with Coriolis effect and the miniaturized optical fiber gyrometric sensors. The analytical modeling of the most common errors and of the most important ones was followed in order to finally have the possibility to implement and generate software without loading their subroutines with unnecessary loopholes, leads to major delays and even jams when numerically simulating a landing control system in the complete. It is extremely important in this modeling that it was considered to be easy to decouple the error channels between them, so that when implementing software and including models in landing simulation schemes, one can study both the effects of each type of error, and the effects of overlapping different types of errors on the performance of aircraft landing control systems.

For accelerometer sensors, bias errors, non-linearity of scale factor, sensitivity axis non-alignment, sensitivity to accelerations applied perpendicular to the input axis, calibration of scale factor and noise were considered in the model. Referring in particular to new systems based on the use of miniaturized sensors, it is noteworthy that the sensor technical datasheets contain statistically deduced values from testing a limited number of units that characterize a very large batch. Therefore, the modeling proceeds from the values of the deterministic errors provided by the technical datasheets, to which are added different deviations within the limit deduced from the specifications in the same sheets. Generation of noise, superimposed on the other errors, is done according to its density and the frequency band specified in the modeling sheet technical sheet. The same modeling procedure is applied to the noise density value as in the case of the values of the deterministic errors provided by the technical data sheets.

As there is a strong variation in temperature over the relatively short time span of the landing of the aircraft, the influence of temperature on the bias and the scale factor has been modeled. The temperature parameter is entered indirectly in the analytical error model of the sensor, derived from the standard atmospheric computation relations, taking into account the atmospheric layers and the altitude flight value. A polynomial modeling of temperature influences has been chosen, which is customized according to the information in the data sheet characteristic of the type of sensor to be modeled.

In the modeling of the girometers, there were considered errors due to bias, scale factor calibration, scale factor nonlinearity, sensitivity to accelerations applied by an arbitrary direction, sensitivity to angular speeds applied perpendicular to the input and noise axis. Similar to the modeling of accelerometers, various deviations from the noise density value and the deterministic error values provided by the datasheets were considered. Also, the developed model took into account the temperature influences on bias and scale factor.

Activity I.4. Design of multiple observers for longitudinal and lateral motions of aircraft during landing

The fourth of the five activities of Stage I aims at designing a multiple observer structure to allow estimation of aircraft state during the landing process. The multiple observer to be obtained must have a generality of character so that it can be used both for longitudinal and side-to-side aircraft movements during the three main landing stages. Activity I.4 therefore involves the achievement of ***OS4*** (*Design of a multiple observer for aircraft landing in longitudinal plane*) and ***OS6*** (*Design of a multiple observer for aircraft landing in lateral-directional plane*).

Below, there is presented the design approach of a multiple observer, the block diagram associated to this observer, the resulting synthetic algorithm, and the software validation of the algorithm for a numerical example associated with the longitudinal motion of aircraft during landing.

The structure of the multiple observer

One considers the multiple model described by the equations [29]

$$\dot{x}(t)=\sum_{i=1}^{M}\mu_i(\xi(t))[A_i x(t)+B_i u(t)+D_i v(t)],\ y(t)=Cx(t)+Fv(t), \tag{1.15}$$

where $x(t)\in\mathcal{M}^{n\times1}$ is the state of the system, $u(t)\in\mathcal{M}^{m\times1}$ – the vector of known inputs, $v(t)\in\mathcal{M}^{q\times1}$ – the vector of unknown inputs, $y(t)\in\mathcal{M}^{p\times1}$ – the vector of outputs, while $C\in\mathcal{M}^{p\times n}$ is the output matrix of the system. For the local model „i”, $A_i\in\mathcal{M}^{n\times n}$ is the matrix associated to the state, $B_i\in\mathcal{M}^{n\times m}$ – the input matrix, $D_i\in\mathcal{M}^{n\times q}$ şi $F\in\mathcal{M}^{p\times q}$ – the transmission matrix. The activation functions $\mu_i(\xi(t)), i=\overline{1,M}$, have the properties $\sum_{i=1}^{M}\mu_i(\xi(t))=1, 0\le\mu_i(\xi(t))\le 1, (\forall)\, i=\overline{1,M}$; the vector $\xi(t)$ is called the decision vector and depends on the inputs/measurable variables. The number of local models (*M*) depends on the precision of modeling, the complexity of the nonlinear system to be approximated by using a multiple model, and the structure of the activation functions. Below, the time dependency (*t*) is omitted.

The design of the multiple observer

One considers the multiple observer [29]

$$\dot{z}=\sum_{i=1}^{M}\mu_i(\xi)(N_i z+G_i u+L_i y), \hat{x}=z-Ey, \tag{1.16}$$

where $N_i\in\mathcal{M}^{n\times n}, G_i\in\mathcal{M}^{n\times m}, L_i\in\mathcal{M}^{n\times p}$ are the gain matrices of the local observer „*i*”, while $E\in\mathcal{M}^{n\times p}$ – transformation matrices. The matrices N_i, G_i, L_i and *E* should be determined such that the estimated state vector $\hat{x}$ tends to x.

One considers the error of the observer $e=x-\hat{x}$, expressed using the equations (1.15), (1.16) and the notation $Q=I+EC$, as follows: $e=x-\hat{x}=x-z+E(Cx+Fv)=x-z+ECx+EFv=Qx-z+EFv$. The dynamics of the error is [29]:

$$\dot{e}=\sum_{i=1}^{M}\mu_i(\xi)[N_i e+(QA_i-N_i-K_iC)x+(QB_i-G_i)u+(QD_i-K_iF)v]+EF\dot{v}, \tag{1.17}$$

where K_i have the forms $K_i=N_iE+L_i$. If there are fulfilled the conditions:

$$QD_i=K_iF, G_i=QB_i, N_i=QA_i-K_iC, EF=0, \tag{1.18}$$

the error has the dynamics: $\dot{e}=\sum_{i=1}^{M}\mu_i(\xi)N_i e$ and tends to zero if there exists a symmetrical and positive defined matrix *P* such that, $(\forall)i=\overline{1,M}$, the inequality $N_i^T P+PN_i<0$ is verified. Thus, the constraints (1.18) and $N_i^T P+PN_i<0$ lead to the complete determination of the multiple observer; additionally, the matrix *F* should be full column rank, i.e. $\text{rang}(F)<p$ [29]. The synthesis of the constraints and equations can be summarized into the following theorem:

<u>*Theorem*</u> [29]

The estimation error between the multiple model (1.15) and the observer (1.16) converges asimptoticaly to zero if there are the matrices *P*>0, *S* and W_i such that, $(\forall)i=\overline{1,M}$, there are fulfilled the conditions:

$$A_i^T P+PA_i+A_i^T C^T S^T+SCA_i-W_iC-C^T W_i^T<0, (P+SC)D_i=W_iF, SF=0. \tag{1.19}$$

In these circumstances, the observer (1.16) is defined by means of the matrices:

$$E=P^{-1}S, G_i=(I+P^{-1}SC)B_i, N_i=(I+P^{-1}SC)A_i-P^{-1}W_iC, L_i=P^{-1}W_i-N_iE. \tag{1.20}$$

<u>*Proof*</u>

The nonlinear inequality $N_i^T P+PN_i<0$, by replacing N_i with the third expression (1.18), using the notations [29]:

$$W_i=PK_i, S=PE, \tag{1.21}$$

becomes the LMI (1.19) – inequality with the unknown matrices *P* and W_i; the matrices A_i and *C* are known, while the matrix *S* is determined from the third condition (1.19) with respect to the matrix *F*. Thus, if the number of columns associated to the matrix *C* is equal to the number of lines associated to the matrix *F*, the matrix *S* is obtained with [30]: $S=I-F(CF)^{+}C$; an alternative method is the usega of the method from [31]. After the determination of the matrices *P*, *S* and W_i (which have to verify the second equation (1.19)), one calculates K_i from the first equation (1.21), written under the form: $K_i=P^{-1}W_i$, and, then, the matrix *E* using the second equation (1.21).

The second equation (1.19) is obtained by left multiplying with P the first equation (1.18), while the third equation (1.19) is obtained by left multiplying with P the fourth equation (1.18). Thus, the eqs. (1.19) and the first eq. (1.20) have been proved. The obtaining of the three equations (1.20) is achieved by replacing in (1.18) Q with $I+EC$, matrix E with $P^{-1}S$, matrices K_i with $K_i = P^{-1}W_i$ and taking into account the equation $Q = I + EC$. For the solving of the LMI (1.19), one uses the Schur lemma written under the form: $\begin{bmatrix} \widetilde{Q} & \widetilde{S} \\ \widetilde{S}^T & \widetilde{R} \end{bmatrix} < 0$ if and only if $\widetilde{R} < 0$ şi $\widetilde{Q} - \widetilde{S}\widetilde{R}^{-1}\widetilde{S}^T < 0$; the Schur lemma has been particularized as follows: $\widetilde{Q} = A_i^T P + PA_i + A_i^T C^T S^T + SCA_i - W_i C - C^T W_i^T$, $\widetilde{S} = 0$, $\widetilde{R} = -I$. The block diagram for the equations (1.15) and (1.16) is presented in fig. 1.4.

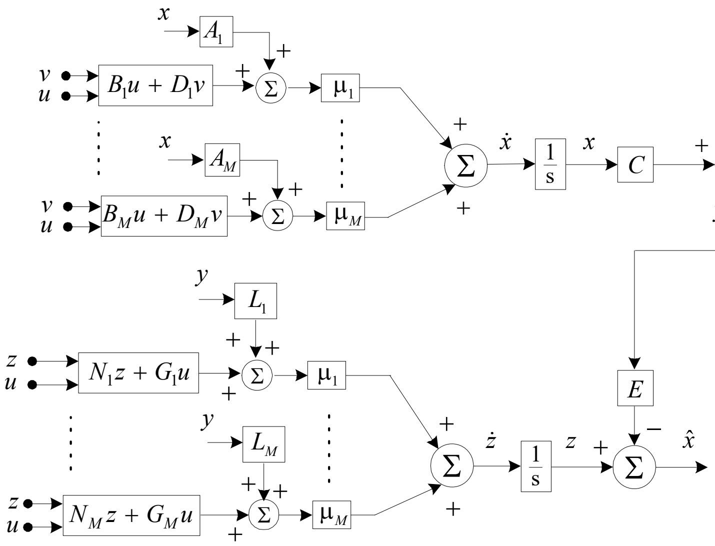


Fig. 1.4. Block diagram of the Akhenak multiple observer

The estimation of the unknown inputs

On ehas proved that the observer (1.16) is convergent if the condition from the previous theorem are satisfied. In stationary regime, the error e tends to zero; replacing x with $\hat{x}$ in eqs. (1.15), one gets [29]:

$$\begin{cases} \dot{\hat{x}} = \sum_{i=1}^{M} \mu_i(\xi)\left(A_i\hat{x} + B_iu + D_i\hat{v}\right), \\ \hat{y} = C\hat{x} + F\hat{v}, \end{cases} \tag{1.22}$$

wheree $\hat{v}$ is the estimation of the vector associated to the unknown inputs (v), while $\hat{y}$ is the estimation of the system's output (y). $\hat{v}$ is calculated in [29] as following:

$$\hat{v} = \left(W^TW\right)^{-1}W^T \begin{bmatrix} \dot{\hat{x}} - \sum_{i=1}^{M} \mu_i(\xi)\left(A_i\hat{x} + B_iu\right) \\ \hat{y} - C\hat{x} \end{bmatrix}, \tag{1.23}$$

where

$$W = \begin{bmatrix} \dot{\hat{x}} - \sum_{i=1}^{M} \mu_i(\xi)D_i \\ F \end{bmatrix}; \tag{1.24}$$

W should be full column rank matrix in order to calculate $W^+ = \left(W^TW\right)^{-1}W^T$. If F is full column rank matrix, the estimation of the unknown input vector is completed as below [29]:

$$\hat{v} = \left(F^T F\right)^{-1} F^T \left(y - \hat{y}\right). \tag{1.25}$$

Validation of the multiple observer

The observer is validated in Matlab/Simulink; for this validation, one chooses the longitudinal motion of an aircraft; its dynamics is described by eqs. (1.16) where:

$$A_1 = \begin{bmatrix} -0.007 & 0.012 & -9.81 & 0 \\ -0.128 & -0.54 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.065 & 0.96 & 0 & -0.99 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ -0.04 \\ 0 \\ -12.5 \end{bmatrix}, D_1 = \begin{bmatrix} 0.07 \\ 0.02 \\ 0.07 \\ 0.08 \end{bmatrix}, A_2 = \begin{bmatrix} -0.01 & 0.01 & -9.81 & 0 \\ -0.128 & -0.6 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.1 & 1.2 & 0 & -1.5 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0.1 \\ -0.1 \\ 0 \\ -10.8 \end{bmatrix}, D_2 = \begin{bmatrix} 0.05 \\ 0.01 \\ 0.06 \\ 0.07 \end{bmatrix}, x^T = \left[\Delta V_x \quad \Delta\alpha \quad \Delta\theta \quad \Delta\omega_y\right], u = \delta_p, C = I_4, F = \left[0.01 \quad 0 \quad 0.01 \quad 0\right]^T.$$

The vector of unknown inputs has been randomly used; for the above matrices, one obtained: $M = 2, n = p = 4, m = q = 1$. From the aircraft dynamics' point of view, the longitudinal or vertical wind shears, atmospheric turbulences, or errors of the sensors represent unknown inputs; an observer for systems with unknown inputs may estimate these unknown inputs, but, more important, it should estimate the system's states with very small errors [4]; therefore, the unknown input vector has been randomly chosen. The decision and the activation functions are: $\xi(t) = u(t), \mu_1(\xi(t)) = 0.5\left(1 - \tanh\left(\xi(t)\right)\right), \mu_2(\xi(t)) = 0.5\left(1 + \tanh\left(\xi(t)\right)\right)$. One obtained the graphics from figs. 1.5 and 1.6.

The simulation is achieved for random input (fig. 1.5) and an input of form $u = -K\hat{x}$ (fig. 1.6), where K is determined with an optimal algorithm; one used the ALGLX optimal algorithm from [9]. În fig. 1.5 one represented the estimation errors of the multiple observer (case 1: random input; in fig. 1.6 there are represented the system's state variables $x_i(t)-$ continous line and the estimated state variables $\hat{x}_i(t)-$ dashed line.

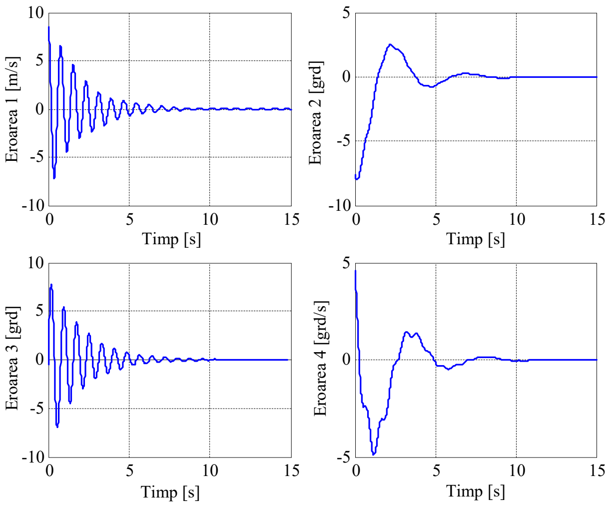


Fig. 1.5. Estimation errors of the observer (random input)

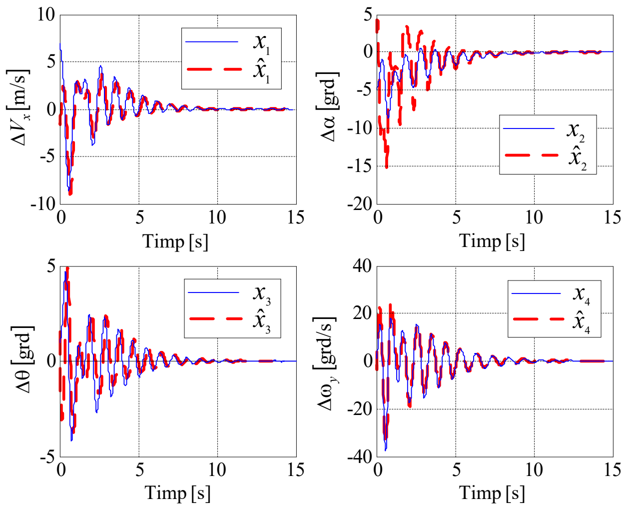


Fig. 1.6. State variables and the estimated state variables

Activity I.5. Design of the web page associated to the project – dissemination of the results

The activity was scheduled to meet the sixth objective of the stage. This has been carried out throughout the whole phase and has been accomplished through the creation of a web page (http://elth.ucv.ro/site/mlungu/index.php?language=ro&page=48) – Romanian language and in English language (http://elth.ucv.ro/site/mlungu/index.php?language=en&page=50), which reflects the current state of the project.

Designing sensory analytical models taking into account both their deterministic and stochastic errors, the design of a multiple sighting for landing aircraft in the longitudinal and lateral-directional planes, as well as a good documentation of landing geometry, landing aircraft dynamics, errors sensors, atmospheric disturbances occurring during landing, and the adaptive and optimal methods currently in place to control landing aircraft allowed the research team to develop 3 articles addressing the issues outlined above. Two of the articles were submitted for review and publication to 2 ISI journals (Journal of Dynamic Systems, Measurement and Control - Transactions of ASME and Neurocomputing Journal) and an IEEE Conference (17th IEEE International Charpathian Control Conference, 26 May - 1 June, 2016, Slovakia). Below there are presented the titles of the 3 articles, as well as a small summary of these papers.

- Lungu, R., Lungu, M., *Design of Automatic Landing Systems using the H-inf Control and the Dynamic Inversion.* Journal of Dynamic Systems, Measurement and Control (Transactions of ASME), DOI:10.1115/1.4032028**,** ISSN: 0022-0434, e-ISSN: 1528-9028 (**ISI Journal**).

The paper focuses on the automatic control of aircraft in the longitudinal plane, during landing, by using the linearized dynamics of aircraft, taking into consideration the wind shears and the errors of the sensors. A new robust automatic landing system is obtained by means of the H-inf control, the dynamic inversion, an optimal observer and two reference models providing the aircraft desired velocity and altitude. The theoretical results are validated by numerical simulations for a Boeing 747 landing; the simulation results are very good (Federal Aviation Administration accuracy requirements for Category III are met) and show the robustness of the system even in the presence of wind shears and sensor errors. Moreover, the designed control law has the ability to reject the sensor measurement noises and wind shears with low intensity.

- Lungu, R., Lungu, M., *Adaptive Flight Control Law Based on Neural Networks and Dynamic Inversion for Micro Aerial Vehicles.* Neurocomputing Journal, vol. 171, pp. 471-481, 2016, ISSN: 0925-2312 **(ISI Journal)**.

The paper presents two new adaptive systems, for the attitude‘s control of the micro aerial vehicles (MAVs) – insect type. The dynamic model describing the motion of MAVs with respect to the Earth tied frame is nonlinear and the design of the new adaptive control system is based on the dynamic inversion technique. The inversion error is calculated with respect to the control law and two matrices (inertia and dynamic damping matrices) which express the deviation of the estimated matrices relative to the calculated ones (the matrices from the nonlinear dynamics of MAVs) in conditions of absolute stability in closed loop system by using the Lyapunov theory. To completely compensate this error, an adaptive component (output of a neural network) is added in the control law. The system also includes a second order reference model which provides the desired attitude vector and its derivative. The two variants of the new adaptive control system are validated by complex numerical simulations.

- Lungu, R., Lungu, M., Tutunea, D., *The Control of Aircraft Landing using the Dynamic Inversion and the H-inf Control*. 17th IEEE International Charpathian Control Conference (**ISI Proceedings**), 26 May - 1 June, 2016, Slovacia.

The paper focuses on the automatic control of aircraft in the longitudinal plane, during landing, by using the literalized dynamics of aircraft, taking into consideration the wind shears and the errors of the sensors. The H-inf control provides robust stability with respect to the uncertainties caused by different disturbances and noise type signals, while the dynamic inversion provides good precision tracking. A new robust automatic landing system (ALS) is obtained by means of the H-inf control, the dynamic inversion, optimal observers, a dynamic compensator and two reference models providing the aircraft desired velocity and altitude. The theoretical results are validated using numerical simulations for a light aircraft landing; the simulation results are very good (Federal Aviation Administration accuracy requirements for Category III are met) and show the robustness of the algorithm even in the presence of wind shears and sensor errors.

It is also worth mentioning that part of the information obtained from this stage of the project was used for the improvement of the course *Automatic flight flight control* from the Master of Education at the University of Craiova.

Taking into account the full accomplishment of Stage I activities: Documentation on aircraft landing and design of analytical models of multiple sensors and observers, we consider that all 5 specific activities of the stage were fully met and all 6 specific objectives of the stage I have been accomplished. The fulfillment of all the objectives initially set at this stage creates the premises for successfully solving the next stage of the project, "***Design of the optimal ALS; Validation and Optimization***", Phase II – 2016.

References (phase I of the project)

[1] Donald, Mc.L. *Automatic Flight Control Systems.* New York, London, Toronto, Sydney, Tokyo, Singapore, 1990.

[2] Shue, S., Agarwal, R.K., Design of automatic landing systems using mixed H_2/H_∞ control, Journal of Guidance, Control, and Dynamics, vol. 22, pp. 103-114, 1999.

[3] Liao, F., Wang, J.L., Poh, E.K. *Fault – Tolerant Robust Automatic Landing Control Design*, Journal of Guidance, Control and Dynamics, vol. 28, nr. 5, 2005, pp. 854-871.

[4] Juang, J.G., Cheng, K.C. *Application of Neural Network to Disturbances Encountered Landing Control*. IEEE Transactions on Intelligent Transportation Systems, vol. 7, nr. 4, 2006, pp. 582-588.

[5] Juang, J.G., Lee, C.L. *Application of Cerebellar Model Articulation controllers to Inteligent Landing System*. Journal of Universal Computer Science, vol. 15, nr. 13, 2009, pp. 2586-2607.

[6] *** U.S. Military Specification MIL – 8785C, 1980.

[7] Che, J., Chen, D. *Automatic Landing Control using H-inf control and Stable Inversion*. Proceedings of the 40th Conference on Decision and Control, Orlando, Florida, USA, 2001, pp. 241-246.

[8] Che, J., Chen, D. *Automatic Landing Control using H-inf control and Stable Inversion*, Proceedings of the 40th Conference on Decision and Control, Orlando, Florida, USA, 2001, pp. 241-246.

[9] Lungu, M. *Sisteme de conducere a zborului (Flight control systems)*, Sitech Publisher, 2008.

[10] Pashilkar, A., Sundararajan, N., Saratchhandran, P.A. *Fault-Tolerant Neural Aided controller for Aircraft Auto-Landing*, Aerospace Science and Technology, vol. 10, nr. 1, 2006, pp. 49-61.

[11] Shau-Shiun, J., Gebre-Egziabher, D., Walter, T., Enge, P. *Improving GPS-based landing system performance using an empirical barometric altimeter confidence bound*, IEEE Transactions on Aerospace and Electronic Systems, vol. 44, nr. 1, 2008, pp. 127-146.

[12] Yicheng, L., Zhang, T., Jingyan, S., Khan, M.J. *Controller design for high-order descriptor linear systems based on requirements on tracking performance and disturbance rejection*, Aerospace Science and Technology, vol. 13, nr. 7, 2009, pp. 364-373.

[13] Li, Y., Sundararajan, N., Saratchandran, P., Wang, Z. *Robust Neuro-H_∞ controller design for aircraft auto-landing*, IEEE Transactions on Aerospace and Electronic Systems, vol. 40, nr. 1, 2004, pp. 158-167.

[14] Mori, R., Suzuki, S. *Neural Network Modeling of Lateral Pilot Landing Control*, Journal of Aircraft, vol. 46, 2009, pp. 1721-1726.

[15] Juang, J., Chang, H., Chang, W. *Intelligent automatic landing system using time delay neural network controller*, Applied Artificial Intelligence: An International Journal, vol. 17, nr. 7, 2003, pp. 563-581.

[16] Vo, H., Sridhar, S. *Robust Control of F-16 Lateral Dynamics*, International Journal of Aerospace and Mechanical Engineering 2008, pp. 80-85.

[17] Venkateswara, D.M., Tiauw, H.G. *Automatic landing system design using sliding mode control*, Aerospace Science and Technology, vol. 32, nr. 1, 2014, pp. 180-187.

[18] Zdenko, K., Stjepan, B. *Fuzzy Controller Design – Theory and applications*, Taylor and Francis, 2006.

[19] Kuma, V., Rana, K.P., Gupt, V. *Real-Time Performance Evaluation of a Fuzzy PI + Fuzzy PD Controller for Liquid-Level Process*, International Journal of Intelligent Control and Systems, vol. 13, nr. 2, 2008, pp. 89-96.

[20] Prasad, B., Pradeep, S. *Automatic Landing System Design using Feedback Linearization Method*, AIAA 2007-2733, AIAA Infotech@Aerospace Conference and Exhibit, Rohnert Park, California, 2007.

[21] Singh, S., Padhi, R., *Automatic Path Planning and Control Design for Autonomous Landing of UAVs using Dynamic Inversion*, American Control Conference Riverfront, St. Louis, MO, USA , 2009, pp. 2409-2414.

[22] Ochi, Y., Kanai, K. *Automatic approach and landing for propulsion controlled aircraft by H_∞ control*. IEEE International Conference on Control Applications, Hawaii, 1999, pp. 997-1002.

[23] Hui, S., Zak, S. *Observer design for systems with unknown inputs*, Int. J. appl. Math. Comput. Sci., vol. 15, nr. 4, 2005, pp.

431-446.

[24] Lungu, M., Lungu, R. *Design of linear functional observers for linear systems with unknown inputs*, International Journal of Control, vol. 85, nr. 10, 2012, pp. 1602-1615.

[25] Bhattacharyya, S.P. *Observer design for linear systems with unknown inputs*, IEEE Trans. Automat. Contr., vol. AC-23, 1978, pp. 483-484.

[26] Chadli, M., Akhenak, A., Ragot, J., Maquin, D. *State and Unknown Input Estimation for Discrete Time Multiple Model*, Journal of the Franklin Institute, vol. 346, nr. 6, 2009, pp. 593-610.

[27] Koenig, D., Marx, B., Jacquet, D. *Unknown input observers for switched nonlinear discrete time descriptor systems*, IEEE Transactions on Automatic Control, vol. 53, 2008, pp. 373-379.

[28] Boubaker, O. *Robust Observers for Linear Systems with Unknown Inputs: a Review*, ASCE Journal, vol. 5, 2005, pp. 45-51.

[29] Akhenak, A., Chadli, M., Ragot, J., Maquin, D., *Unknown input multiple obser-ver based-approach - application to secure communications*. 1st IFAC Conference on analysis and Control of Chaotic Systems, Reims, France, 2006.

[30] Akhenak, A., Chadli, M., Maquin, D., Ragot, J., *State estimation via multiple observer. Three tank system*. 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Washington, 2003, pp. 1227-1232.

[31] Trinh, H., Ha, Q., *Design of linear functional observers for linear systems with unknown inputs*. International Journal of Systems Science, vol. 31, nr. 6, 2000, pp. 741-749.

SYSNTHESYS OF THE WORK,

containing the activity and the obtained results in comparison with the objectives of the project

for

- Stage II (2016) -

Stage II (2016), ***Design, validation and optimizing of the optimal Auto Landing System***, took 12 months (January - December); in this period, one has completely performed all the **8 activities**:

II.1. Design of the H_2/H_∞ control laws (longitudinal and lateral-directional planes).

II.2. Design of the blocks for the reference models, geometry of landing and the dynamic compensator (longitudinal and lateral-directional planes).

II.3. Design of the PCH blocks using classical or fuzzy methods (longitudinal and lateral-directional planes).

II.4. Design of the optimal landing control systems (longitudinal and lateral-directional planes).

II.5. Interconections of the two optimal subsystems and the obtaining of a new system for the control of aircraft landing.

II.6. Software implementation of the system for the landing optimal control.

II.7. Organizing of a special session within IEEE International Conference - ICATE'16.

II.8. Results' dissemination.

and one has also accomplished all the **6 specific objectives:**

OS5. To design a new optimal landing system (longitudinal plane) by using the H_2/H_∞ technique, the dynamic inversion approach, reference models, the geometry of landing, and dynamic compensators, taking into account the sensor errors, the wind shears, and the atmospheric turbulences (*innovative architecture*).

OS8. To interconnect the two automatic landing subsystems and to obtain a new and innovative ALS based on the H_2/H_∞ technique and the dynamic inversion approach (*innovative auto-pilot architecture*).

OS9. To software implement, test and validate the new optimal ALS (*software package*).

OS12. To design a Pseudo Control Hedging block (PCH), for aircraft motion in longitudinal plane, by using a new method based on fuzzy logic (*innovative architecture*).

OS21. To disseminate the results in the scientific, academic and socio-economic environment.

Beside the acomplish of the scientific activities, the members of **the team achieved administrative and management activities (elaboration of scientific rapports, tasks distribution, tracking deadlines etc.), which competed at the completion of this stage in good condition.** Also, there were **regular meetings** between team members of the team, especially given that the two PhD students involved in the project are PhD students of "Politehnica" University of Bucharest. Also, the research team was met in full in eight meetings to analyze activities and results so far achieved and to determine future actions of each member program. Funds allocated at this stage both for mobility, logistic and staff costs were managed successfully so that all objectives of the stage have been achieved.

Below, it is presented the synthetic work done and results achieved in each of the eight activities intended to be conducted at this stage.

Activity II.1. Design of the H_2/H_∞ control laws (longitudinal and lateral-directional planes).

Design of the control law (longitudinal plane)

Aircraft dynamics in two planes has been deduced within *Activity I.1. Documentary studies concerning aircraft dynamics, the error sensors, and the atmospheric disturbances during landing* (2015).

Landing is one of the most critical stages of flight; the aircraft has to perform a precise maneuver in the proximity of the ground to land safely at a suitable touch point with acceptable sink rate, speed, and attitude. During aircraft landing, the presence of different unknown or partially known disturbances in aircraft dynamics leads to the necessity of using modern automatic control systems. Sometimes, the conventional controllers are difficult to use due to the drastically changing of the atmospheric conditions and the dynamics of aircraft [7], [8]. In order to control aircraft landing, the feedback linearization has been used in [9], but the drawback of this method is that all the parametric plant uncertainties must appear in the same equation of the state-space representation. Other automatic landing systems use feed-forward neural networks based on the back propagation learning algorithms [10]; the main disadvantage is that the neural networks require a priori training on normal and faulty operating data. From the optimal synthesis' point of view, a mixed technique for the H_2/H_∞ control of landing has been introduced by Shue and Agarwal [11], while Ochi and Kanai [12] have used the H-inf technique for the same purpose; the negative point of these papers is the robustness of the controllers since the sensor errors and other external disturbances are not considered. The fuzzy logic has been also used to imitate the pilot's experience in compromising between trajectory tracking and touchdown safety [13]. In other studies [14], there has been proved that an intelligent on-line-learning controller is helpful in assisting different baseline controllers in tolerating a stuck control surface in the presence of strong wind.

The control of aircraft during the two landing phases is achieved in this section by designing a controller which uses both the H_2/H_∞ control and the dynamic inversion technique. The aim of the H_2 optimal control method is to improve the overshoot of the dynamic processes, while the H_∞ technique is a very good choice for the minimization of the disturbances effect on the system output variables (imposed as performance variables); thus, in this subsection, it will be obtained a mixed H_2/H_∞ control law which will be the first component $(\hat{\boldsymbol{u}})$ of aircraft's control law ($\boldsymbol{u}$).

To design the H_2/H_∞ control law, we choose the following output vector: $z = [z_1 \quad z_2]^T = [\theta \quad u]^T$; the equations associated to the output variables to be controlled (aircraft pitch angle - θ and the longitudinal velocity - u) are:

$$z_1 = C_0\boldsymbol{x} + D_{01}\boldsymbol{u},\ z_2 = C_1\boldsymbol{x} + D_{11}\boldsymbol{u}, \tag{2.1}$$

For glide slope, taking into account the system's state equation and $\boldsymbol{x} = \begin{bmatrix} \frac{u}{V_0} & \alpha & q & \theta & \frac{H}{V_0} & \frac{\dot{H}}{V_0} & \delta_e & \delta_T \end{bmatrix}^T$, one yields:

$$C_0 = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0], C_1 = [V_0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0], D_{01} = [c_1 \quad 0], D_{11} = [0 \quad c_2]; \tag{2.2}$$

c_1 and c_2 are positive constants. Moreover, the equation of the measurement system can be written of the following form:

$$y = C\boldsymbol{x} + D_{22}e, \tag{2.3}$$

where $D_{22} = \bar{k}I_7$ ($\bar{k}$ – positive constant) is the matrix of weights associated to the vector containing the sensor errors [15]: $e = [e_u \quad e_\alpha \quad e_q \quad e_\gamma \quad e_H \quad e_{\delta_e} \quad e_{\delta_T}]^T, y = [u \quad \alpha \quad q \quad \gamma \quad H \quad \delta_p \quad \delta_T]^T$, while the matrix C is obtained by identification.

For the second phase of landing (flare), the equations (3) remain valid, but the forms of the vectors e and y become now: $e = [e_u \quad e_\alpha \quad e_q \quad e_\theta \quad e_H \quad e_{\delta_e} \quad e_{\delta_T}]^T, y = [u \quad \alpha \quad q \quad \theta \quad H \quad \delta_p \quad \delta_T]^T$; D_{22} has the same expression as the one in the case of the first landing phase, while the matrix C can be again easily obtained by identification.

Putting together the equation $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + G\boldsymbol{u}_w$ and the equations (2.1) and (2.3), the following equation is obtained:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} A_{(8\times8)} & B_{(8\times2)} & G_{(8\times3)} & 0_{(8\times7)} \\ C_{0(1\times8)} & D_{01(1\times2)} & 0_{(1\times3)} & 0_{(1\times7)} \\ C_{1(1\times8)} & D_{11(1\times2)} & 0_{(1\times3)} & 0_{(1\times7)} \\ C_{(7\times8)} & 0_{(7\times2)} & 0_{(7\times3)} & D_{22(7\times7)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ \boldsymbol{u_w} \\ e \end{bmatrix}. \tag{2.4}$$

To proof that, in steady regime, the forms of $z_1 = \theta$ and $z_2 = u$ are the same with the ones in equation (2.1) and (2.4), we used the expansion of $z = [z_1 \quad z_2]^T$ as function of $\boldsymbol{x}$ and $\boldsymbol{u}$; for $\boldsymbol{u}_0$=0, we have successively obtained the following equations:

$$z=\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}=z(\boldsymbol{x},\boldsymbol{u})\cong \underbrace{z(\boldsymbol{x}_0,\boldsymbol{u}_0)}_{z_0}+\left(\frac{\partial z}{\partial \boldsymbol{x}}\right)_{(x_0,0)}\Delta\boldsymbol{x}+\left(\frac{\partial z}{\partial \boldsymbol{u}}\right)_{(x_0,0)}\Delta\boldsymbol{u}\cong z_0+\underbrace{\begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_n} \\ \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_2}{\partial x_n} \end{bmatrix}_{(x_0,0)}}_{\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}}\Delta\boldsymbol{x}+\underbrace{\begin{bmatrix} \frac{\partial z_1}{\partial u_1} & \frac{\partial z_1}{\partial u_2} \\ \frac{\partial z_2}{\partial u_1} & \frac{\partial z_2}{\partial u_2} \end{bmatrix}_{(x_0,0)}}_{\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}}\Delta\boldsymbol{u}\Leftrightarrow \Delta z\cong\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}\Delta\boldsymbol{x}+\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}\Delta\boldsymbol{u}\Leftrightarrow$$

$z\cong\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}\boldsymbol{x}+\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}\boldsymbol{u}$, where $C_0=\begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_n} \end{bmatrix}_{(x_0,0)}$, $C_1=\begin{bmatrix} \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_2}{\partial x_n} \end{bmatrix}_{(x_0,0)}$, $x_i\left(i=\overline{1,8}\right)$ are the system's states, $u_i\left(i=\overline{1,2}\right)$ are the system's inputs $(\delta_{ec},\delta_{Tc})$, $D_{01}=\begin{bmatrix} \frac{\partial z_1}{\partial u_1} & \frac{\partial z_1}{\partial u_2} \end{bmatrix}_{(x_0,0)}=\begin{bmatrix} \frac{\partial \theta}{\partial \delta_{e_c}} & \frac{\partial \theta}{\partial \delta_{T_c}} \end{bmatrix}_{(x_0,0)}=[c_1 \quad 0]$, $D_{11}=\begin{bmatrix} \frac{\partial z_2}{\partial u_1} & \frac{\partial z_2}{\partial u_2} \end{bmatrix}_{(x_0,0)}=\begin{bmatrix} \frac{\partial u}{\partial \delta_{e_c}} & \frac{\partial u}{\partial \delta_{T_c}} \end{bmatrix}_{(x_0,0)}=[0 \quad c_2]$. ■

The control law $\boldsymbol{u}$ is calculated by using the formula: $\boldsymbol{u}=\hat{\boldsymbol{u}}+\overline{\boldsymbol{u}}$, where its first component $(\hat{\boldsymbol{u}})$ is the optimal command calculated with the H_2/H_∞ method, while $\overline{\boldsymbol{u}}$ is determined by using the dynamic inversion. For obtaining $\hat{\boldsymbol{u}}$ (output of H_2/H_∞ controller), one has to minimize two cost functions. The first one, associated to the H_2 approach, is: $J_1=\frac{1}{2}\int_0^\infty z_1^T z_1 \mathrm{d}t=\frac{1}{2}\int_0^\infty\left[\boldsymbol{x}^T\underbrace{\left(C_0^T C_0\right)}_{Q_0}\boldsymbol{x}+\hat{\boldsymbol{u}}^T\underbrace{\left(D_{01}^T D_{01}\right)}_{R_0}\hat{\boldsymbol{u}}\right]\mathrm{d}t$.

In the case of H_2 approach, the optimal control law has been determined in [16] as: $\hat{\boldsymbol{u}}=-K\hat{\boldsymbol{x}}$, $K=R_0^+B^TP, R_0=D_{01}^TD_{01}$; here, R_0^+ denotes the pseudo-inverse of the matrix R_0, the symmetric and positive defined matrix $P\in R^{8\times 8}$ is the stabilizing solution of the matrix Riccati equation [15]:

$$A^TP+PA-PBR_0^+B^TP+Q_0=0, \tag{2.5}$$

while $\hat{\boldsymbol{x}}$ is the aircraft estimated state (obtained by means of an observer which will be designed by using the H_2 approach); denoting with $\Delta\boldsymbol{x}$ - the difference (deviation) between the aircraft real state ($\boldsymbol{x}$) state and its desired value $(\overline{\boldsymbol{x}})$, i.e. $\Delta\boldsymbol{x}=\boldsymbol{x}-\overline{\boldsymbol{x}}$, by means of the observer borrowed from [16], i.e.:

$$\Delta\dot{\hat{\boldsymbol{x}}}=A\Delta\hat{\boldsymbol{x}}+B\boldsymbol{u}+G\boldsymbol{u}_{\boldsymbol{w}}+L\left(\Delta y-C\Delta\hat{\boldsymbol{x}}\right), \tag{2.6}$$

we obtain the estimation of the signal $\Delta\boldsymbol{x}$, i.e. $\Delta\hat{\boldsymbol{x}}=\hat{\boldsymbol{x}}-\overline{\boldsymbol{x}}$; the observer gain matrix L is calculated as: $L=P^*C^T\left(D_{22}^TD_{22}\right)^{-1}$, with P^*- the stabilizing solution of the matrix Riccati equation [15]:

$$AP^*+P^*A^T-P^*C^TCP^*+GG^T=0; \tag{2.7}$$

to design the observer (2.1), the vector of disturbances $\boldsymbol{u}_w$ has been estimated by the equipment from aircraft's navigation system.

The H_∞ control method combines the classical shaping and the notion of bandwidth with modern H_∞ robust stabilization; by using now this control approach, the system state equation, the equation of the output variable z_2, and the equation of the system measurement equation, one calculates, for the two landing stages, the optimal control law $\hat{\boldsymbol{u}}$ which minimizes the cost function: $J_2=\frac{1}{2}\int_0^\infty z_2^Tz_2\mathrm{d}t=\frac{1}{2}\int_0^\infty\left[\boldsymbol{x}^T\underbrace{\left(C_1^TC_1\right)}_{Q_1}\boldsymbol{x}+\hat{\boldsymbol{u}}^T\underbrace{\left(D_{11}^TD_{11}\right)}_{R_1}\hat{\boldsymbol{u}}\right]\mathrm{d}t$. As in the previous case (H_2 approach), the form of the optimal control law has been also deduced in [16]: $\hat{\boldsymbol{u}}=-K_\infty\hat{\boldsymbol{x}}, K_\infty=R_1^+B^TP_\infty, R_1=D_{11}^TD_{11}$; R_1^+ denotes the pseudo-inverse of the matrix R_1, while the symmetric and positive defined matrix $P_\infty\in R^{8\times 8}$ is the solution of the matrix Riccati equation [15], [16]:

$$A^TP_\infty+P_\infty A-P_\infty\left(BR_1^+B^T-\mu_1^{-2}GG^T\right)P_\infty+Q_1=0. \tag{2.8}$$

Here, Q_1 and R_1 are positive matrices, while μ_1 is a small enough positive scalar for which the Riccati equation (2.8) has a stabilizing solution. The observer equation is again (2.1) with the gain matrix having the form: $L=L_\infty=P_\infty^*C^T\left(D_{22}^TD_{22}\right)^{-1}\in R^{8\times 7}$, where $P_\infty^*\in R^{8\times 8}$ is the stabilizing solution of the matrix Riccati equation:

$$AP_\infty^* + P_\infty^* A^T - P_\infty^*\left(C^T C - \mu_2^{-2} C_1^T C_1\right)P_\infty^* + GG^T = 0; \tag{2.9}$$

μ_2 has the same significance in (2.9) as μ_1 in (2.8), but these two constants can have the same values in some particular cases [15].

In order to obtain an optimum for the control law calculated both with H_2 and H_∞ approaches, i.e. to obtain the expression of the H_2/H_∞ control law, the following algorithm can be used:

Optimal algorithm [15]:

Step 1: The determination of the norms H_2 and H_∞ associated to the solutions of the Riccati equations (2.5), (2.7), (2.8), and (2.9);

Step 2: The calculation of the matrices $\hat{P} = (1-k)P + kP_\infty$, $\hat{P}^* = (1-k)P^* + kP_\infty^*$, with $k \in (0,1)$;

Step 3: The check of the following conditions' fulfillment: $\hat{P}, \hat{P}^* \geq 0, \hat{P}\hat{P}^* < I$.

Step 4: If the above conditions are not met, one chooses again the constant k and the steps 2 and 3 are again run;

Step 5: The final expression of the H_2/H_∞ control law is obtained with the equation: $\hat{\boldsymbol{u}} = -\hat{K}\hat{\boldsymbol{x}}, \hat{K} = R_1^+ B^T \hat{P}$, while the final expression of the observer gain matrix is $\hat{L} = \hat{P}^* C^T \left(D_{22}^T D_{22}\right)^{-1}$. ■

We consider the vector $\bar{z} = \begin{bmatrix}\bar{\theta} & \bar{u}\end{bmatrix}^T$ containing the reference variables (the desired values of the pitch angle and longitudinal velocity, respectively). By using $\bar{z}$ and the dynamic inversion principle, we calculate $\bar{\boldsymbol{x}}$ (the desired state of the system) and $\bar{\boldsymbol{u}}$ (the desired control) with respect to $\bar{z}$ and, after that, the vector $\bar{y}$ is obtained with the equations: $\dot{\bar{\boldsymbol{x}}} = A\bar{\boldsymbol{x}} + B\bar{\boldsymbol{u}}$, $\bar{z} = C'\bar{\boldsymbol{x}}, \bar{y} = C\bar{\boldsymbol{x}}; C' = \begin{bmatrix} C_0 \\ C_1 \end{bmatrix}$, with C_0 and C_1 having the forms (2.6). Now, a coordinates' change is achieved by means of the transformation matrix $T \in R^{8\times 8}$ [17]:

$$\begin{bmatrix}\xi \\ \eta\end{bmatrix} = T\boldsymbol{x}, \boldsymbol{x} = T^{-1}\begin{bmatrix}\xi \\ \eta\end{bmatrix}, \xi = \begin{bmatrix} z_1 & \dot{z}_1 & \cdots & z_1^{(r_1-1)} & z_2 & \dot{z}_2 & \cdots & z_2^{(r_2-1)} & \cdots & z_p & \dot{z}_p & \cdots & z_p^{(r_p-1)}\end{bmatrix}^T;$$

ξ is a state vector consisting of the controlled variables and their derivates, with $z_i^{(r_i-1)}$ – the (r_i-1) order derivative of z_i; for the aircraft dynamics in longitudinal plane, $z_1=\theta$, $z_2=z_p=u$. The second state vector (η) contains the states which are not included in the vector ξ; the dimension of vector η is $n - r = n - \sum_{i=1}^{p} r_i$, where n is the dimension of the square matrix T, while the values of r_i are deduced below. Thus, one derivates with respect to time the equations of z_1 and z_2 until the terms containing the components of the control law $(\delta_{ec}, \delta_{Tc})$ appear; one yields:

$$\begin{aligned}
\dddot{\theta} &= a'_{41}\frac{u}{V_0} + a'_{42}\alpha + a'_{43}q + a'_{44}\theta + a'_{47}\delta_e + a'_{48}\delta_T + \frac{b_{31}}{T_e}\delta_{ec} + \frac{b_{32}}{T_T}\delta_{Tc} + a'_{41}\frac{u_g}{V_0} + a'_{42}\alpha_g + a'_{43}q_g + V_0 a_{31}\frac{\dot{u}_g}{V_0} + V_0 a_{33}\dot{\alpha}_g + a_{33}\dot{q}_g, \\
\ddot{u} &= a'_{11}\frac{u}{V_0} + a'_{12}\alpha + a'_{13}q + a'_{14}\theta + a'_{17}\delta_e + a'_{18}\delta_T + \frac{b_{11}}{T_e}\delta_{ec} + \frac{b_{12}}{T_T}\delta_{Tc} + g'_{11}\frac{u_g}{V_0} + g'_{12}\alpha_g + g'_{13}q_g + g_{11}\frac{\dot{u}_g}{V_0} + g_{12}\dot{\alpha}_g,
\end{aligned} \tag{2.10}$$

with

$$\begin{aligned}
&a'_{41} = V_0\left(a_{11}a_{31} + a_{21}a_{33} + a_{31}a_{33}\right), a'_{42} = V_0\left(a_{12}a_{31} + a_{22}a_{33} + a_{31}a_{33}\right), a'_{43} = a_{23}a_{33} + a_{33}^2 + a_{34}, \\
&a'_{44} = a_{31}a_{14}, a'_{47} = a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} - \frac{b_{31}}{T_e}, a'_{48} = a_{31}b_{12} + a_{33}b_{22} + a_{33}b_{32} - \frac{b_{32}}{T_T}; \\
&a'_{11} = V_0\left(a_{11}^2 + a_{12}a_{21}\right), a'_{12} = V_0\left(a_{11}a_{12} + a_{12}a_{22}\right), a'_{13} = a_{12}a_{23} + a_{14}, a'_{14} = a_{11}a_{14}, a'_{17} = a_{11}b_{11} + a_{12}b_{21} - \frac{b_{11}}{T_e}, \\
&a'_{18} = a_{11}b_{12} + a_{12}b_{21} - \frac{b_{12}}{T_T}, g'_{11} = V_0\left(a_{11}^2 + a_{12}a_{21}\right), g'_{12} = V_0\left(a_{11}a_{12} + a_{12}a_{22}\right), g'_{13} = a_{14}, g_{11} = V_0 a_{11}, g_{12} = V_0 a_{12}.
\end{aligned} \tag{2.11}$$

According to (2.10), the relative degrees are $r_1=3$ and $r_2=2$ and the equations (2.10) may be combined into a single one:

$z^{(r)} = A_x \boldsymbol{x} + B_u \overline{\boldsymbol{u}} + G' \boldsymbol{u}_w + G'' \dot{\boldsymbol{u}}_w$, with $z^{(r)} = \begin{bmatrix} z_1^{(r_1)} & z_2^{(r_2)} \end{bmatrix}^T = \begin{bmatrix} \ddot{\theta} & \ddot{u} \end{bmatrix}^T, \overline{\boldsymbol{u}} = \begin{bmatrix} \delta_{ec} & \delta_{Tc} \end{bmatrix}^T, A_x = \begin{bmatrix} a'_{41} & a'_{42} & a'_{43} & a'_{44} & 0 & 0 & a'_{47} & a'_{48} \\ a'_{11} & a'_{12} & a'_{13} & a'_{14} & 0 & 0 & a'_{17} & a'_{18} \end{bmatrix}, B_u = \begin{bmatrix} \frac{b_{31}}{T_e} & \frac{b_{32}}{T_T} \\ \frac{b_{11}}{T_e} & \frac{b_{12}}{T_T} \end{bmatrix}$,

$G' = \begin{bmatrix} a'_{41} & a'_{42} & a'_{43} \\ g'_{11} & g'_{12} & g'_{13} \end{bmatrix}, G'' = \begin{bmatrix} V_0 a_{31} & V_0 a_{33} & a_{33} \\ g_{11} & g_{12} & 0 \end{bmatrix}$. For the glide slope phase, the state vectors ξ and η are: $\xi = \begin{bmatrix} \theta & \dot{\theta} & \ddot{\theta} & u & \dot{u} \end{bmatrix}^T, \eta = \begin{bmatrix} \frac{H}{V_0} & \frac{\dot{H}}{V_0} & \delta_e \end{bmatrix}^T$, while, for the second phase of landing, these vectors are, respectively: $\xi = \begin{bmatrix} \theta & \dot{\theta} & \ddot{\theta} & u & \dot{u} \end{bmatrix}^T, \eta = \begin{bmatrix} \frac{H}{V_0} & \frac{\widetilde{H}_c}{V_0} & \delta_e \end{bmatrix}^T$. Using the coordinates' change [17], considering $\boldsymbol{u}_w = 0$ and $\boldsymbol{u} = \overline{\boldsymbol{u}}$, the system gets the form: $\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \end{bmatrix} = \hat{A} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \hat{B} \overline{\boldsymbol{u}}$; $\hat{A} = TAT^{-1}, \hat{B} = TB$. If the matrices $\hat{A}$ and $\hat{B}$ are partitioned with respect to the dimensions of vectors ξ and η, it results [17]: $\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} \overline{\boldsymbol{u}}$; the matrices in the above equation have the following dimensions: $\hat{A}_{11} \in R^{5\times5}, \hat{A}_{12} \in R^{5\times3}, \hat{A}_{21} \in R^{3\times5}, \hat{A}_{22} \in R^{3\times3}, \hat{B}_1 \in R^{5\times2}, \hat{B}_2 \in R^{3\times2}$. The matrix A has the form (4) for the glide slope phase and the form (2.6) for the second landing phase; as a consequence, the matrices $\hat{A}$ and $\hat{A}_{ij}, i, j = \overline{1,2}$, modify during the transition from the first landing phase to the second one. One also obtains: $\dot{\xi} = \hat{A}_{11}\xi + \hat{A}_{12}\eta + \hat{B}_1 \overline{\boldsymbol{u}}, \dot{\eta} = \hat{A}_{21}\xi + \hat{A}_{22}\eta + \hat{B}_2 \overline{\boldsymbol{u}}$. Imposing $\xi = \overline{\xi}$ and $\dot{\xi} = \dot{\overline{\xi}}$, with $\overline{\xi} = \begin{bmatrix} \overline{\theta} & \dot{\overline{\theta}} & \ddot{\overline{\theta}} & \overline{u} & \dot{\overline{u}} \end{bmatrix}^T, \dot{\overline{\xi}} = \begin{bmatrix} \dot{\overline{\theta}} & \ddot{\overline{\theta}} & \dddot{\overline{\theta}} & \dot{\overline{u}} & \ddot{\overline{u}} \end{bmatrix}^T$, from equation $\dot{\xi} = \hat{A}_{11}\xi + \hat{A}_{12}\eta + \hat{B}_1 \overline{\boldsymbol{u}}, \dot{\eta} = \hat{A}_{21}\xi + \hat{A}_{22}\eta + \hat{B}_2 \overline{\boldsymbol{u}}$, the vector $\overline{\boldsymbol{u}}$ results:

$$\overline{\boldsymbol{u}} = \hat{B}_1^+ \left(\dot{\overline{\xi}} - \hat{A}_{11}\overline{\xi} - \hat{A}_{12}\eta \right). \tag{2.12}$$

For the glide slope, one obtains:

$$\begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \\ u \\ \dot{u} \\ \frac{H}{V_0} \\ \frac{\dot{H}}{V_0} \\ \delta_e \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ V_0 a_{31} & V_0 a_{32} & a_{33} & a_{34} & 0 & 0 & b_{31} & b_{32} \\ V_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ V_0 a_{11} & V_0 a_{12} & 0 & a_{14} & 0 & 0 & b_{11} & b_{12} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{T} \begin{bmatrix} \frac{u}{V_0} \\ \alpha \\ q \\ \theta \\ \frac{H}{V_0} \\ \frac{\dot{H}}{V_0} \\ \delta_e \\ \delta_T \end{bmatrix}; \tag{2.13}$$

for the second landing phase, in (2.13) $\frac{\dot{H}}{V_0}$ is replaced by $\frac{\widetilde{H}_c}{V_0}$. Now, replacing $\overline{\xi}$ and $\dot{\overline{\xi}}$ in (2.12), one gets:

$$\overline{\boldsymbol{u}} = \hat{B}_1^+ \left\{ \begin{bmatrix} 0 \\ 0 \\ \dddot{\overline{\theta}} \\ 0 \\ \ddot{\overline{u}} \end{bmatrix} - \begin{bmatrix} \hat{a}_{11} & (\hat{a}_{12} - 1) & \hat{a}_{13} & \hat{a}_{14} & \hat{a}_{15} \\ \hat{a}_{21} & \hat{a}_{22} & (\hat{a}_{23} - 1) & \hat{a}_{24} & \hat{a}_{25} \\ \hat{a}_{31} & \hat{a}_{32} & \hat{a}_{33} & \hat{a}_{34} & \hat{a}_{35} \\ \hat{a}_{41} & \hat{a}_{42} & \hat{a}_{43} & \hat{a}_{44} & (\hat{a}_{45} - 1) \\ \hat{a}_{51} & \hat{a}_{52} & \hat{a}_{53} & \hat{a}_{54} & \hat{a}_{55} \end{bmatrix} \overline{\xi} - \hat{A}_{12}\eta \right\} \tag{2.14}$$

or $\overline{\boldsymbol{u}} = \hat{B}_1^+ \left\{ \begin{bmatrix} 0 & 0 & \dddot{\overline{\theta}} & 0 & \ddot{\overline{u}} \end{bmatrix}^T - \hat{A}'_{11}\overline{\xi} - \hat{A}_{12}\eta \right\}$, where $\hat{A}'_{11}$ is calculated from $\hat{A}_{11}$ making the substitutions $\hat{a}'_{12} = \hat{a}_{12} - 1$, $\hat{a}'_{23} = \hat{a}_{23} - 1, \hat{a}'_{45} = \hat{a}_{45} - 1$, the other elements of the matrices $\hat{A}_{11}$ and $\hat{A}'_{11}$ being the same; $\hat{a}_{ij}, i, j = \overline{1,5}$ are the elements of the matrix $\hat{A}_{11}$. Replacing $\dot{\xi} = \hat{A}_{11}\xi + \hat{A}_{12}\eta + \hat{B}_1\overline{\boldsymbol{u}}$, $\dot{\eta} = \hat{A}_{21}\xi + \hat{A}_{22}\eta + \hat{B}_2\overline{\boldsymbol{u}}$ in (2.14), with $\xi = \overline{\xi}$, one successively obtains

$$\dot{\eta} = \hat{A}_{21}\overline{\xi} + \hat{A}_{22}\eta + \hat{B}_2\hat{B}_1^{+}\left\{\begin{bmatrix}0 & 0 & \ddot{\overline{\theta}} & 0 & \ddot{\overline{u}}\end{bmatrix}^T - \hat{A}'_{11}\overline{\xi} - \hat{A}_{12}\eta\right\} = \left(\hat{A}_{22} - \hat{B}_2\hat{B}_1^{+}\hat{A}_{12}\right)\eta + \left(\hat{A}_{21} - \hat{B}_2\hat{B}_1^{+}\hat{A}'_{11}\right)\overline{\xi} + \hat{B}_2\hat{B}_1^{+}\begin{bmatrix}0 & 0 & \ddot{\overline{\theta}} & 0 & \ddot{\overline{u}}\end{bmatrix}^T; \quad (2.15)$$

(2.15) can be expressed as:

$$\dot{\eta} = \hat{A}_{\eta}\,\eta + \hat{B}_z\overline{z}^{(r)} + \hat{A}_{\xi}\overline{\xi} \Leftrightarrow \dot{\eta} = \hat{A}_{\eta}\,\eta + \hat{B}_y\overline{Z}, \tag{2.16}$$

where $\hat{A}_{\eta} = \hat{A}_{22} - \hat{B}_2\hat{B}_1^{+}\hat{A}_{12}, \hat{A}_{\xi} = \hat{A}_{21} - \hat{B}_2\hat{B}_1^{+}\hat{A}'_{11}, \hat{B}_z\overline{z}^{(r)} = \hat{B}_2\hat{B}_1^{+}\begin{bmatrix}0 & 0 & \ddot{\overline{\theta}} & 0 & \ddot{\overline{u}}\end{bmatrix}^T, \overline{z}^{(r)} = \begin{bmatrix}\overline{z}_1^{(r_1)} & \overline{z}_2^{(r_2)}\end{bmatrix}^T = \begin{bmatrix}\ddot{\overline{\theta}} & \ddot{\overline{u}}\end{bmatrix}^T$, $\hat{B}_y = \begin{bmatrix}\hat{B}_z & \hat{A}_{\xi}\end{bmatrix}$,

$\overline{Z} = \begin{bmatrix}\overline{z}^{(r)} & \overline{\xi}\end{bmatrix}^T = \begin{bmatrix}\ddot{\overline{\theta}} & \ddot{\overline{u}} & \overline{\theta} & \dot{\overline{\theta}} & \ddot{\overline{\theta}} & \overline{u} & \dot{\overline{u}}\end{bmatrix}^T$. The dimensions of the above matrices are: $\hat{A}_{\eta} \in R^{3\times3}, \hat{A}_{\xi} \in R^{3\times5}, \hat{B}_z \in R^{2\times2}, \hat{B}_y \in R^{3\times7}$. If

$\hat{B}_2\hat{B}_1^{+} = \begin{bmatrix}\hat{b}_{11} & \hat{b}_{12} & \hat{b}_{13} & \hat{b}_{14} & \hat{b}_{15}\\ \hat{b}_{21} & \hat{b}_{22} & \hat{b}_{23} & \hat{b}_{24} & \hat{b}_{25}\\ \hat{b}_{31} & \hat{b}_{32} & \hat{b}_{33} & \hat{b}_{34} & \hat{b}_{35}\end{bmatrix}$, then $\hat{B}_z^T = \begin{bmatrix}\hat{b}_{13} & \hat{b}_{23} & \hat{b}_{33}\\ \hat{b}_{15} & \hat{b}_{25} & \hat{b}_{35}\end{bmatrix}$. Thus, for the calculation of vector $\overline{\boldsymbol{u}}$, one solves equation (2.16) – second form and obtains the vector η and then uses equation (2.14). From the expression of $\hat{B}_z\overline{z}^{(r)}$, it results: $\hat{B}_1^{+}\begin{bmatrix}0 & 0 & \ddot{\overline{\theta}} & 0 & \ddot{\overline{u}}\end{bmatrix}^T = \hat{B}_2^{+}\hat{B}_z\overline{z}^{(r)}$, which, replaced into (2.14), leads to the following one:

$$\overline{\boldsymbol{u}} = \hat{B}_u^{-1}\left(\overline{z}^{(r)} - \hat{B}_{\xi}\overline{\xi} - \hat{B}_{\eta}\eta\right), \tag{2.17}$$

with $\hat{B}_u^{-1} = \hat{B}_2^{+}\hat{B}_z, \hat{B}_{\xi} = \hat{B}_u\hat{B}_1^{+}\hat{A}'_{11}, \hat{B}_{\eta} = \hat{B}_u\hat{B}_1^{+}\hat{A}_{12}$; these matrices have the following dimensions: $\hat{B}_u \in R^{2\times2}, \hat{B}_{\xi} \in R^{2\times5}, \hat{B}_{\eta} \in R^{2\times3}$. Therefore, $\overline{\boldsymbol{u}}$ can be obtained by means of equation (2.14) or by using equation (2.17).

Another form of the command law $\overline{\boldsymbol{u}}$ results from the equation $z^{(r)} = A_x\boldsymbol{x} + B_u\overline{\boldsymbol{u}} + G'\boldsymbol{u}_w + G''\dot{\boldsymbol{u}}_w$, if we impose the convergence of $z^{(r)} = \begin{bmatrix}\ddot{\theta} & \ddot{u}\end{bmatrix}^T$ to $\overline{z}^{(r)} = \begin{bmatrix}\ddot{\overline{\theta}} & \ddot{\overline{u}}\end{bmatrix}^T$ and the convergence of the system estimated state $(\hat{\boldsymbol{x}})$ to $\boldsymbol{x}$; in these conditions, it yields:

$$\overline{\boldsymbol{u}} = \hat{B}_u^{-1}\left(\overline{z}^{(r)} - A_x\hat{\boldsymbol{x}} - G'\boldsymbol{u}_w - G''\dot{\boldsymbol{u}}_w\right), \tag{2.18}$$

with $\hat{B}_u, A_x, G'$ and G'' having the same forms as above.

Design of the control law (lateral-directiona plane)

One showed above that in the expressions of $\hat{P}$ and $\hat{P}^*$ the constant *k* represents the weight of the matrices P_∞ and P_∞^*, while the constant (1-*k*) represents the weight of the matrices *P* and P^* in the matrices $\hat{P}$ and $\hat{P}^*$ calculated by means of the H_2/H_∞ technique. With other words, the constant *k* may be regarded as the weight of the H_∞ technique in the control law design, while the constant (1-*k*) may be regarded as the weight of the H_2 technique in the control law design. If *k*=1, the control law is designed only by using the H_∞ technique, while the control law is designed only by using the H_2 technique if *k*=0. Because in lateral-directional plane the desired trajectory tracking errors may be larger than in longitudinal plane and because the improving of the dynamic processes' overshoot (the role of the H_2 optimal control method) is not important, we consider in the design of the control law for lateral-directional plane that *k*=1 (the control law will be achieved using the H_∞ technique).

We start the design of the control law *u* considering that a flying object's nonlinear dynamics, with one input (*u*) and one output (*z*), is generally described by the equations [18]: $\dot{x} = f(x,\overline{u}), z = h(x)$, with $x(n\times1)$ – the state vector, *n* – the number of the state variables, *f* and *h* – nonlinear functions, generally unknown, *u* and *z* – the system's input and output vectors, respectively. We assume that *h*(*x*) is a uniquely invertible function. This dynamics satisfies the hypothesis [19]: $z^{(\mathrm{r})} = h_{\mathrm{r}}(x,\overline{u}), h_{\mathrm{r}} = \dfrac{\mathrm{d^r}h}{\mathrm{d}t^{\mathrm{r}}} = h^{(\mathrm{r})}$,

$\frac{\partial h_i}{\partial u}=0, 0\le i\le r, \frac{\partial h_r}{\partial u}\ne 0$; this means that all derivatives $z^{(i)}=h^{(i)}(x,u)=h^{(i)}, i=\overline{0,r}$, do not depend on $\bar{u}$, while the derivative $z^{(r)}=h_r(x,\bar{u})=h^{(r)}$ depends on $\bar{u}$; r is the relative degree of the system, while z and $z^{(r)}$ are $z=[Y \quad \beta]^T=[h_1(x) \quad h_2(x)]^T$ and $z^{(r)}=\left[z_1^{(r_1)} \quad z_2^{(r_2)}\right]^T=[\ddot{Y} \quad \dot{\beta}]^T=\left[h_1^{(r_1)}(x) \quad h_2^{(r_2)}(x)\right]^T$, respectively; also, $r=r_1+r_2$. Denoting with $\hat{h}_r(z,\bar{u})$– the best approximation of the function $h_r(x,u)=h_r(x(z),\bar{u})=h_r(z,\bar{u})$, we design the control law (the pseudo-command): $\hat{v}=h_r(z,\hat{u})$. The previous equations are equivalent with the following ones: $\bar{u}=h_r^{-1}(z,v), \hat{u}=h_r^{-1}(z,\hat{v})$; if $\hat{h}_r=h_r$, then, one gets $z^{(r)}=v=\hat{v}$; otherwise, $z^{(r)}=\hat{v}+\varepsilon$, where $\varepsilon=\varepsilon(x,\bar{u})=h_r(z,\bar{u})-\hat{h}_r(z,\hat{u})$ is the approximation error of the function h_r (the inversion error), which acts like a disturbing signal of the system. Moreover, the previous equation expresses the presence of r ideal integrators between the pseudo-command (pseudo-control) v and the output z. Imposing that $z\to\bar{z}, \dot{z}\to\dot{\bar{z}}, \ldots, z^{(r)}\to\bar{z}^{(r)}$, the signal v can be chosen of form [20]: $v=\hat{v}+\varepsilon=\bar{z}^{(r)}+\hat{v}_{pd}+\varepsilon$; $\hat{v}_{pd}$ is the output of the linear dynamic compensator, used for the control of the linear subsystem. The system $z^{(r)}=\hat{v}+\varepsilon$, having the input $v=\hat{v}+\varepsilon$ and the output z, is described by the transfer matrix $H_d(s)=\mathrm{diag}\{H_{d_1}(s), H_{d_2}(s)\}$, with $H_{d_i}(s), i=\overline{1,2}$, of forms: $H_{d_i}(s)=\frac{b_{0i}}{s^{r_i}+\lambda_{r_i-1,i}s^{r_i-1}+\ldots+\lambda_{1,i}s+\lambda_{0i}}$. Using the notations: $Z=\left[z \quad \dot{z} \quad \ldots \quad z^{(r-1)}\right]^T=[z_1 \quad \dot{z}_1 \quad \ddot{z}_1 \quad z_2 \quad \dot{z}_2]^T=$ $=[Y \quad \dot{Y} \quad \ddot{Y} \quad \beta \quad \dot{\beta}]^T, \lambda=\begin{bmatrix}\lambda_1 & 0_{r_1\times 1}\\ 0_{r_2\times 1} & \lambda_2\end{bmatrix}=\begin{bmatrix}[\lambda_{01} \quad \lambda_{11} \quad \ldots \quad \lambda_{r_i-1,1}]^T & 0_{r_1\times 1}\\ 0_{r_2\times 1} & [\lambda_{02} \quad \lambda_{12} \quad \ldots \quad \lambda_{r_2-1,2}]^T\end{bmatrix}$, the sub-system $z^{(r)}=\hat{v}+\varepsilon$ is described by equation: $z^{(r)}+\lambda^T Z=b_0(\hat{v}+\varepsilon), b_0=\mathrm{diag}\{b_{01}, b_{02}\}$; if $z^{(r)}=\bar{z}^{(r)}$ and $Z=\bar{Z}$, the error $\varepsilon=[\varepsilon_1 \quad \varepsilon_2]^T$ becomes $\varepsilon=0_{2\times 1}$; for $b_0=I_2, \lambda_{01}=\lambda_{11}=\lambda_{02}=\lambda_{12}=0$, the following equation results: $\hat{v}_r=\bar{z}^{(r)}+\lambda^T\bar{Z}=\left[\dddot{\bar{Y}}+\lambda_{21}\ddot{\bar{Y}} \quad \ddot{\bar{\beta}}\right]^T$. For $z_1=Y$ and $z_2=\beta$, choosing the transfer functions $H_{d_1}(s)=\frac{b_{01}}{s^2(s+\lambda_{21})}, H_{d_2}(s)=\frac{b_{02}}{s^2}$, there will be obtained null stationary errors $(Y_{st}=\dot{Y}_{st}=\beta_{st}=\dot{\beta}_{st})$.

Now, taking into account the equation $z^{(r)}=A_x x+B_u\bar{u}+G'\tilde{w}$, the term $z^{(r)}+\lambda^T Z$ becomes:

$$z^{(r)}+\lambda^T Z=\begin{bmatrix}\dddot{Y}+\ddot{Y}\\ \ddot{\beta}\end{bmatrix}=\begin{bmatrix}a''_{61} & a''_{62} & a''_{63} & a''_{64} & 0 & 0 & a''_{67} & a''_{68}\\ a'_{11} & a'_{12} & a'_{13} & a'_{14} & 0 & 0 & a'_{17} & a'_{18}\end{bmatrix}\hat{x}+\begin{bmatrix}-\frac{V_0 b_{11}}{T_a} & -\frac{V_0 b_{12}}{T_r}\\ \frac{b_{11}}{T_a} & \frac{b_{12}}{T_r}\end{bmatrix}\hat{u}+\begin{bmatrix}a''_{31} & -a_{11}\\ -\frac{a'_{11}}{V_0} & \frac{a_{11}}{V_0}\end{bmatrix}\begin{bmatrix}w\\ \dot{w}\end{bmatrix}, \tag{2.19}$$

where, one has chosen $\lambda_{21}=b_{01}=1$; $\hat{x}$ is the system estimated state, $a''_{61}=a''_{61}-V_0a_{11}, a''_{62}=a'_{62}-V_0a_{12}, a''_{63}=a'_{63}-V_0(a_{13}+1)$, $a''_{64}=a'_{64}-V_0a_{14}, a''_{67}=a'_{67}-V_0b_{11}, a''_{68}=a'_{68}-V_0b_{12}$, and $a''_{31}=a'_{31}+1$. Using the equations $z^{(r)}+\lambda^T Z=b_0(\hat{v}+\varepsilon)$ and (2.19), identifying the terms of the pseudo-control $\hat{v}$ and error ε, we get:

$$\hat{v}=\begin{bmatrix}\hat{v}_1\\ \hat{v}_2\end{bmatrix}=\begin{bmatrix}\hat{h}_{r_1}(z,\hat{u})\\ \hat{h}_{r_2}(z,\hat{u})\end{bmatrix}=\hat{h}_r(\hat{Y},\hat{\beta},\hat{u}), \hat{u}=\left[\hat{\delta}_{a_c} \quad \hat{\delta}_{r_c}\right]^T \Leftrightarrow \hat{v}=\begin{bmatrix}a''_{61} & 0\\ 0 & a'_{11}\end{bmatrix}\begin{bmatrix}\hat{Y}\\ \hat{\beta}\end{bmatrix}+\begin{bmatrix}-\frac{V_0 b_{11}}{T_a} & -\frac{V_0 b_{12}}{T_r}\\ \frac{b_{11}}{T_a} & \frac{b_{12}}{T_r}\end{bmatrix}\begin{bmatrix}\hat{\delta}_{a_c}\\ \hat{\delta}_{r_c}\end{bmatrix}\Leftrightarrow$$
$$\Leftrightarrow \hat{u}=\begin{bmatrix}\hat{\delta}_{a_c}\\ \hat{\delta}_{r_c}\end{bmatrix}=\hat{h}_r^{-1}(\hat{z},\hat{v})=\begin{bmatrix}-\frac{V_0 b_{11}}{T_a} & -\frac{V_0 b_{12}}{T_r}\\ \frac{b_{11}}{T_a} & \frac{b_{12}}{T_r}\end{bmatrix}^{-1}\left(\hat{v}-\begin{bmatrix}a''_{61} & 0\\ 0 & a'_{11}\end{bmatrix}\begin{bmatrix}\hat{Y}\\ \hat{\beta}\end{bmatrix}\right) \tag{2.20}$$

and

$$\varepsilon=\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\end{bmatrix}=\begin{bmatrix}a''_{62} & a''_{63} & a''_{64} & a''_{67} & a''_{68}\\ a'_{12} & a'_{13} & a'_{14} & a'_{17} & a'_{18}\end{bmatrix}\left[\hat{p} \quad \hat{r} \quad \hat{\varphi} \quad \hat{\delta}_a \quad \hat{\delta}_r\right]^T+\begin{bmatrix}a''_{31} & -a_{11}\\ -\frac{a'_{11}}{V_0} & \frac{a_{11}}{V_0}\end{bmatrix}\begin{bmatrix}w\\ \dot{w}\end{bmatrix}; \tag{2.21}$$

in the previous equations $\hat{Y},\hat{\beta},\hat{p},\hat{r},\hat{\varphi},\hat{\psi},\hat{\delta}_a$, and $\hat{\delta}_r$ are the components of the estimated state vector $\hat{x}$. Using the Taylor series expansion of the function $\bar{u}=h_{\mathrm{r}}^{-1}(z,v)=\hat{h}_{\mathrm{r}}^{-1}(z,\hat{v})+\frac{\mathrm{d}}{\mathrm{d}v}\left(h_{\mathrm{r}}^{-1}(z,v)\right)_{v=\hat{v}}(v-\hat{v})=\hat{h}_{\mathrm{r}}^{-1}(z,\hat{v})+\frac{\mathrm{d}}{\mathrm{d}\hat{v}}\left(\hat{h}_{\mathrm{r}}^{-1}(z,\hat{v})\right)\varepsilon$ and replacing the vector z with its estimate $(\hat{z})$, according to the equation $\bar{u}=h_{\mathrm{r}}^{-1}(z,v)$, $\hat{u}=h_{\mathrm{r}}^{-1}(z,\hat{v})$, one gets:

$$\bar{u}=\hat{u}+\varepsilon\frac{\mathrm{d}}{\mathrm{d}\hat{v}}\left(h_{\mathrm{r}}^{-1}(z,\hat{v})\right). \tag{2.22}$$

Replacing now $\hat{u}$ (form (2.20)) into equation (2.22), the final form of the control law $\bar{u}$ is obtained as follows:

$$\bar{u}=\begin{bmatrix}\delta_{a_c}\\ \delta_{r_c}\end{bmatrix}=\hat{h}_{\mathrm{r}}^{-1}(\hat{z},v)=\begin{bmatrix}-\dfrac{V_0 b_{11}}{T_a} & -\dfrac{V_0 b_{12}}{T_r}\\ \dfrac{b_{11}}{T_a} & \dfrac{b_{12}}{T_r}\end{bmatrix}^{-1}\left(v-\begin{bmatrix}a_{61}'' & 0\\ 0 & a_{11}'\end{bmatrix}\begin{bmatrix}\hat{Y}\\ \hat{\beta}\end{bmatrix}\right); \tag{2.23}$$

thus, the control law $\bar{u}$ may be directly calculated by means of (2.20), where $\hat{u}$ is replaced by $\bar{u}$ and $\hat{v}$ by $v=\hat{v}+\varepsilon$. The general control law u has again the form $u=\bar{u}+u_\infty$, where the component u_∞ is obtained below by using the H-inf control.

Let us consider the vector $z=\begin{bmatrix}Y & \beta\end{bmatrix}^T=C'x$ containing the system's controllable output variables and the vector $\bar{z}=\begin{bmatrix}\bar{Y} & \bar{\beta}\end{bmatrix}^T$ containing the system's reference variables (the imposed values of aircraft's lateral deviation and sideslip angle). The system's output vector is y, chosen of the form: $y=\begin{bmatrix}Y & \dot{Y} & \beta & \varphi & p & \psi & r\end{bmatrix}^T=Cx$; the sensors' errors have been not taken into account here. The matrices $C\in R^{7\times 8}$ and $C'\in R^{2\times 8}$ are respectively:

$$C=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ -V_0 & 0 & 0 & 0 & V_0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, C'=\begin{bmatrix}0 & 1\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}^T. \tag{2.24}$$

The system's state equation, the equations associated to $z_1=Y$ and $z_2=\beta$, as well as the equation of the output vector y, may be combined into the following equation:

$$\begin{bmatrix}\dot{x}\\ z_1\\ z_2\\ y\end{bmatrix}=\begin{bmatrix}A_{(8\times 8)} & B_{(8\times 2)} & G_{(8\times 1)} & 0_{(8\times 7)}\\ C_{0(1\times 8)} & D_{01(1\times 2)} & 0_{(1\times 1)} & 0_{(1\times 7)}\\ C_{1(1\times 8)} & D_{11(1\times 2)} & 0_{(1\times 1)} & 0_{(1\times 7)}\\ C_{(7\times 8)} & 0_{(7\times 2)} & 0_{(7\times 1)} & D_{22(7\times 7)}\end{bmatrix}\begin{bmatrix}x\\ u\\ w\\ e\end{bmatrix}; \tag{2.25}$$

the matrices A,B,G have the forms (3.1) and $C_0=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}, C_1=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, D_{01}=\begin{bmatrix}c_1 & 0\end{bmatrix}, D_{11}=\begin{bmatrix}0 & c_2\end{bmatrix}$; matrix C has the form given in (2.24), while $D_{22}=I_7$ for the vector containing the sensor errors: $e=\begin{bmatrix}e_Y & e_{\dot{Y}} & e_\beta & e_\varphi & e_p & e_\psi & e_r\end{bmatrix}^T$. It is known that the bias and the noise are the most severe sensor errors during landing. Usually, on aircraft there are used giros to measure the angular rates (in our case p and r); by integration of the obtained values, the roll and yaw angles result. Moreover, on aircraft there are transducers (sensors) for the attack angle and for the sideslip angle (β). Therefore, we considered here the sensor errors for β, p, and r. The general model of the gyro sensor is the one in [4].

To proof that, in steady regime, the forms of $z_1=Y$ and $z_2=\beta$ are the same with the ones in (2.25), one has used the expansion of $z=\begin{bmatrix}z_1 & z_2\end{bmatrix}^T$ as function of state (x) and of the system command vector (u); for $u_0=0$, one succesively obtained:

$$z=\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}=z(x,u)\cong \underbrace{z(x_0,u_0)}_{z_0}+\left(\frac{\partial z}{\partial x}\right)_{(x_0,0)}\Delta x+\left(\frac{\partial z}{\partial u}\right)_{(x_0,0)}\Delta u\cong z_0+\underbrace{\begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_n} \\ \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_2}{\partial x_n} \end{bmatrix}_{(x_0,0)}}_{\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}}\Delta x+\underbrace{\begin{bmatrix} \frac{\partial z_1}{\partial u_1} & \frac{\partial z_1}{\partial u_2} \\ \frac{\partial z_2}{\partial u_1} & \frac{\partial z_2}{\partial u_2} \end{bmatrix}_{(x_0,0)}}_{\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}}\Delta u \Leftrightarrow \Delta z\cong\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}\Delta x+\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}\Delta u$$

$\Leftrightarrow z\cong\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}x+\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}u\cong\overline{C}\,x+\overline{D}u$, where $C_0=\begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_n} \end{bmatrix}_{(x_0,0)}, C_1=\begin{bmatrix} \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_2}{\partial x_n} \end{bmatrix}_{(x_0,0)}$; $x_i\left(i=\overline{1,n}\right)$ are the state of the system (n=8), $D_{01}=\begin{bmatrix} \frac{\partial Y}{\partial \delta_{a_c}} & \frac{\partial Y}{\partial \delta_{r_c}} \end{bmatrix}_{(x_0,0)}=[c_1 \quad 0], D_{11}=\begin{bmatrix} \frac{\partial z_2}{\partial u_1} & \frac{\partial z_2}{\partial u_2} \end{bmatrix}_{(x_0,0)}=\begin{bmatrix} \frac{\partial \beta}{\partial \delta_{a_c}} & \frac{\partial \beta}{\partial \delta_{r_c}} \end{bmatrix}_{(x_0,0)}=[0 \quad c_2], \overline{C}=\begin{bmatrix} C_0 \\ C_1 \end{bmatrix}, \overline{D}=\begin{bmatrix} D_{01} \\ D_{11} \end{bmatrix}$. c_1 and c_2 have small positive values; in steady regime (u=0), one gets z_1=Y and z_2=β.

The optimal control law has the form [4]:

$$u_\infty=-K_\infty(\hat{x}-\overline{x}), K_\infty=R_1^{-1}B^T P_\infty, R_1=\overline{D}^T\overline{D}; \tag{2.26}$$

u_∞ must minimize the cost functional $J=\frac{1}{2}\int_0^\infty z^T z\mathrm{d}t=\frac{1}{2}\int_0^\infty\left[x^T\underbrace{\left(\overline{C}^T\overline{C}\right)}_{Q_1}x+u_\infty^T\underbrace{\left(\overline{D}^T\overline{D}\right)}_{R_1}u_\infty\right]\mathrm{d}t$. The symmetric and positive defined matrix $P_\infty\in R^{8\times 8}$ is the stabilizing solution of the Riccati matriceal equation [15]

$$A^T P_\infty+P_\infty A-P_\infty\left(BR_1^{-1}B^T-\mu_1^{-2}GG^T\right)P_\infty+Q_1=0. \tag{2.27}$$

Here, R_1 must be a positive defined matrix, while μ_1 is a small enough positive scalar for which the Riccati equation (2.27) has a stabilizing solution. The determination of the controller gain matrix (K_∞) is the so-called H-inf control problem. The plant inputs are classified as control inputs and disturbances. The control input (u) is the output of the controller, which becomes the input to the actuators driving the system (aircraft). Disturbances w=V_{vy} and e are called exogenous inputs; the main distinction between the control input and the exogenous inputs is that the controller can not manipulate exogenous inputs. The plant's outputs are also characterized into two groups; the first group is represented by signals that are measured and become inputs of the controller; the second group is represented here by the performance outputs (z_1 and z_2). The optimal control law u_∞ depends on $\Delta\hat{x}=\hat{x}-\overline{x}$, as one can see in (2.26). To obtain this signal, one uses again the observer (2.6); $w=V_{vy}$ is estimated by means of the equipment from the navigation system. The observer gain matrix $L_\infty\in R^{8\times 7}$ is calculated by using the formula: $L_\infty=P_\infty^* C^T\left(D_{22}^T D_{22}\right)^{-1}$, with P_∞^*- the stabilizing solution of the Riccati matriceal equation [15]: $AP_\infty^*+P_\infty^* A^T-P_\infty^*\left(C^T C-\mu_2^{-2}\overline{C}^T\overline{C}\right)P_\infty^*+GG^T=0$; μ_2 is a small enough positive scalar for which the Riccati equation has a stabilizing solution.

Activity II.2. Design of the blocks for the reference models, geometry of landing and the dynamic compensator (longitudinal and lateral-directional planes)

For aircraft landing in longitudinal plane, the geometry of landing is described by equations (4); the first equation (4) is associated to the glide slope phase, while the second equation (4) is for the flare phase. These equation can have been modeled by the block diagram in Fig. 2.1.

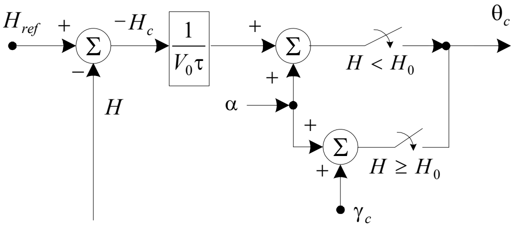


Fig. 2.1. Block diagram for the geometry of landing (longitudinal plane)

For landing in longitudinal plane, two references models are needed. The output of the blocks modeling the reference models is the vector $\overline{Z}$; thus, it may be calculated by means of two reference models, the former being a third order reference model, while the latter is a second order reference model (Fig. 2.2).

For lateral-directional plane, the geometry of landing is described by equations (2.7) and (2.8). Also, the vectors $\bar{z}$ and $\hat{v}_{\mathrm{r}}$ are calculated by means of two reference models; these are presented in Fig. 2.3; $v_{h_1} = v_{h_2} = 0$ (system without PCH block) or $v_{h_1} \neq 0, v_{h_2} \neq 0$ (system considering PCH block). The forms of the signals v_{h_1} and v_{h_2} (the two components of the Pseudo Control Hedging block's output - v_h) will be presented in the following activity.

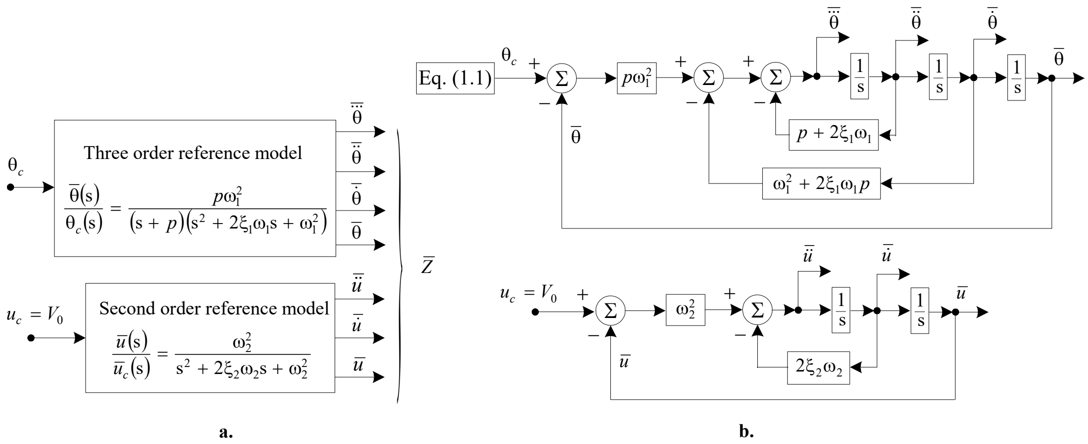


Fig. 2.2. Block diagrams of the three order and second order reference models (longitudinal plane), respectively: a) simplified block diagrams; b) detailed block diagrams

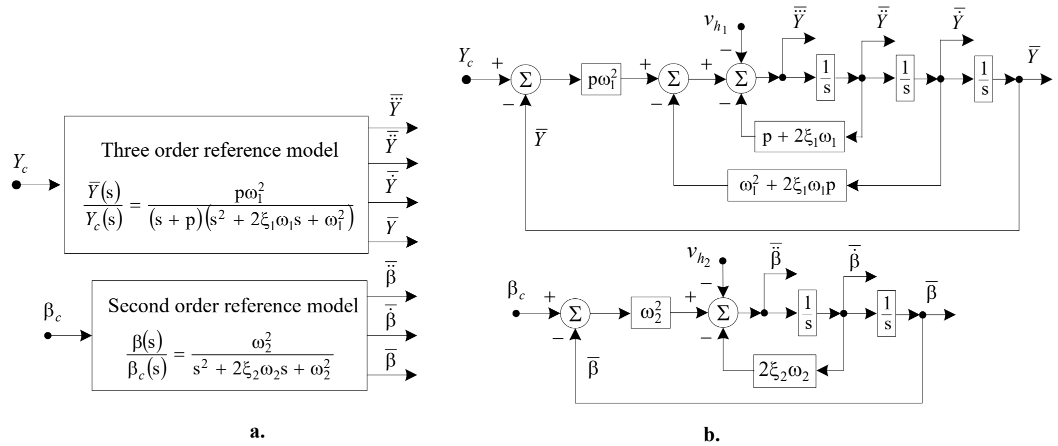


Fig. 2.3. Block diagrams of the three order and second order reference models (lateral-directional plane), respectively: a) simplified block diagrams; b) detailed block diagrams

The linear dynamic compensator for the control of aircraft in lateral-directional plane is chosen as proportional-derivative (PD) one; it provides the signal $\hat{v}_{pd}$ – necessary for the system's stabilization. The proportional coefficients (k_{p_1} and k_{p_2}) and the derivative ones (k_{d_1} and k_{d_2}) associated to the dynamic compensator are calculated by imposing desired roots (solutions) for the characteristic equations of the linear closed loop subsystem with unitary and negative feedback. The two characteristic equations are: $\mathrm{s}^3 + \lambda_{21}\mathrm{s}^2 + b_{01}\mathrm{s} + b_{01}k_{p_1} = 0$ and $\mathrm{s}^2 + \left(b_{02}k_{d_2} + \lambda_{12}\right)\mathrm{s} + b_{02}k_{p2} = 0$, respectively; we have chosen λ_{21}=b_{01}=b_{02}=1 and λ_{12}=0. Thus, the parameters of the dynamic compensator are calculated such that the two characteristic equations presented above have imposed (desired) solutions, i.e. all the complex solutions are placed in the left-hand side of the complex plane. Without loosing the generality, these solutions can be the same with the poles of the reference models' transfer functions.

Activity II.3. Design of PCH blocks using classical or fuzzy methods (longitudinal and lateral planes)

The controllers may be sensitive to actuators' nonlinearities; therefore, in their architectures, one may introduce a block which limits the pseudo-control by means of a component representing an estimation of the execution element's dynamics (PCH - Pseudo Control Hedging) [21]. The input saturation and the input rate saturation may also be significant problems. For lateral-directional plane, we introduce a PCH block which limits the signal v with a component representing the actuator dynamics' estimation. Thus, PCH "moves back" the reference model, introducing a reference model's response correction with respect to the estimation of the execution element's position. The signal provided by PCH $v_h = \begin{bmatrix} v_{h_1} & v_{h_2} \end{bmatrix}^T$ is a reference model's additional input [21]. The block diagrams for the modeling of the servo-ailerons, servo-rudder, and the two components of the PCH block's output (v_{h_1} and v_{h_2}) are presented in Fig. 2.4. The equations that lead to these two block diagrams have resulted by using the equations (3.21). Thus, by means of (3.21), one obtained: $\hat{v}_1 = a''_{61}\hat{Y} - \frac{V_0 b_{11}}{T_a}\hat{\delta}_{a_c} - \frac{V_0 b_{12}}{T_r}\hat{\delta}_{r_c} \cong a''_{61}\hat{Y} - \frac{V_0 b_{11}}{T_a}\hat{\delta}_{a_c} = \hat{h}_{r_1}(z,\hat{u}), \hat{v}_2 = a'_{11}\hat{\beta} + \frac{b_{11}}{T_a}\hat{\delta}_{a_c} + \frac{b_{12}}{T_r}\hat{\delta}_{r_c} \cong a'_{11}\hat{\beta} + \frac{b_{12}}{T_r}\hat{\delta}_{r_c} = \hat{h}_{r_2}(z,\hat{u})$; in these two equations, we considered two independent channels (roll and yaw) and, therefore, we neglected the terms $(-V_0 b_{12}/T_r)\hat{\delta}_{r_c}$ and $(b_{11}/T_a)\hat{\delta}_{a_c}$. It can be easily obtained:

$$\delta_{a_c} \cong \hat{\delta}_{a_c} = \hat{h}_{r_1}^{-1}(\hat{Y},\hat{v}_1) = \frac{T_a}{V_0 b_{11}}\left(a''_{61}\hat{Y} - \hat{v}_1\right),\ \delta_{r_c} \cong \hat{\delta}_{r_c} = \hat{h}_{r_2}^{-1}(\hat{\beta},\hat{v}_2) = \frac{T_r}{b_{12}}\left(-a'_{11}\hat{\beta} + \hat{v}_2\right). \tag{2.28}$$

Using now the above presented equations, with some changes ($\hat{v}_1$ becomes $\bar{v}_1$ and $\hat{v}_2$ becomes $\bar{v}_2$), the following equations have resulted:

$$\bar{v}_1 = a''_{61}\hat{Y} - \frac{V_0 b_{11}}{T_a}\bar{\delta}_a,\ \bar{v}_2 = a'_{11}\hat{\beta} + \frac{b_{12}}{T_r}\bar{\delta}_r. \tag{2.29}$$

The two components of the PCH block's output (v_{h_1} and v_{h_2}) have the forms: $v_{h_1} = \hat{v}_1 - \bar{v}_1$ and $v_{h_2} = \hat{v}_2 - \bar{v}_2$, respectively.

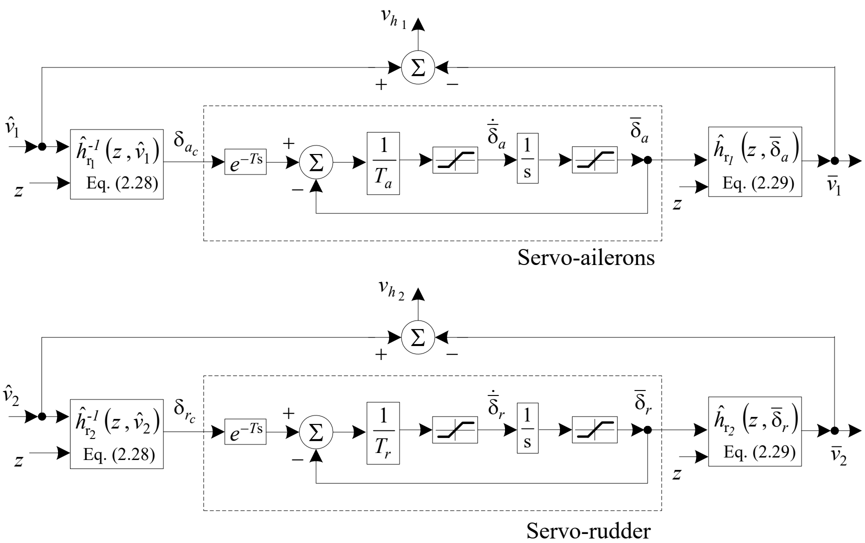


Fig. 2.4. The block diagrams for the modeling of the servo-ailerons, servo-rudder, and the two components of the PCH block's output

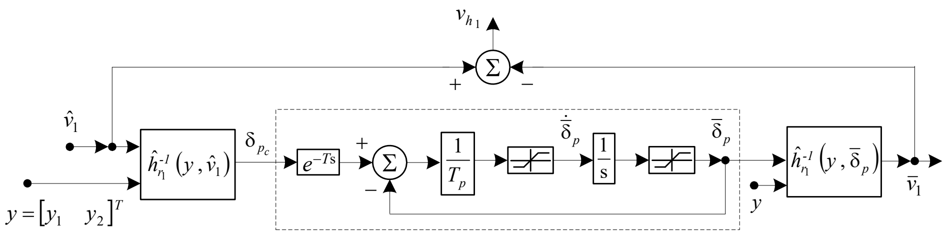


Fig. 2.5. The block diagram for the modeling of the servo-elevator and of the PCH subsystem

The PCH blocks are more useful in the case of neural networks' usage; the strong point of the neural networks is their approximation ability, these being capable to approximate an unknown system dynamics through learning; also, the PCH blocks eliminate the NNs' adapting difficulties. The main purpose of the Pseudo Control Hedging block is to prevent the adaptive element of a control system from trying to adapt to a class of system input characteristics (characteristics of the plant or of the controller). For longitudinal plane, the signal provided by PCH (v_{h_1}) is a reference model's additional input (Fig. 2.5).

Activity II.4. Design of the optimal landing control systems (longitudinal and lateral planes)

The structure of the automatic landing (longitudinal plane) subsystem using the H_2/H_∞ control technique, the dynamic inversion method, two reference models, an optimal observer, and the geometry of landing is presented in Fig. 2.6. Here, the variables with a line above mean the desired (commanded) values of these variables, while the matrix C_r has the form $C_r = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0]$. Taking into account all the above equations, it can be concluded that the automatic control of the aircraft in longitudinal plane, during landing, is mainly based on the dynamic inversion and H_2/H_∞ method. The obtaining of the aircraft desired landing trajectory in longitudinal plane mainly involves two variables' control: the pitch angle (θ) and the forward speed (u). According to the landing requirements for Boeing 747, the aircraft must descend from cruising altitude to a lower altitude around 420 m. Meanwhile, the aircraft speed is also reduced from the cruising speed to an approach value and, after that, it remains constant. So, when we design the desired trajectory, we design the desired forward speed u first of all. The optimal control system associated to aircraft flight during landing (longitudinal plane), based on H_2/H_∞ and dynamic inversion techniques, and the

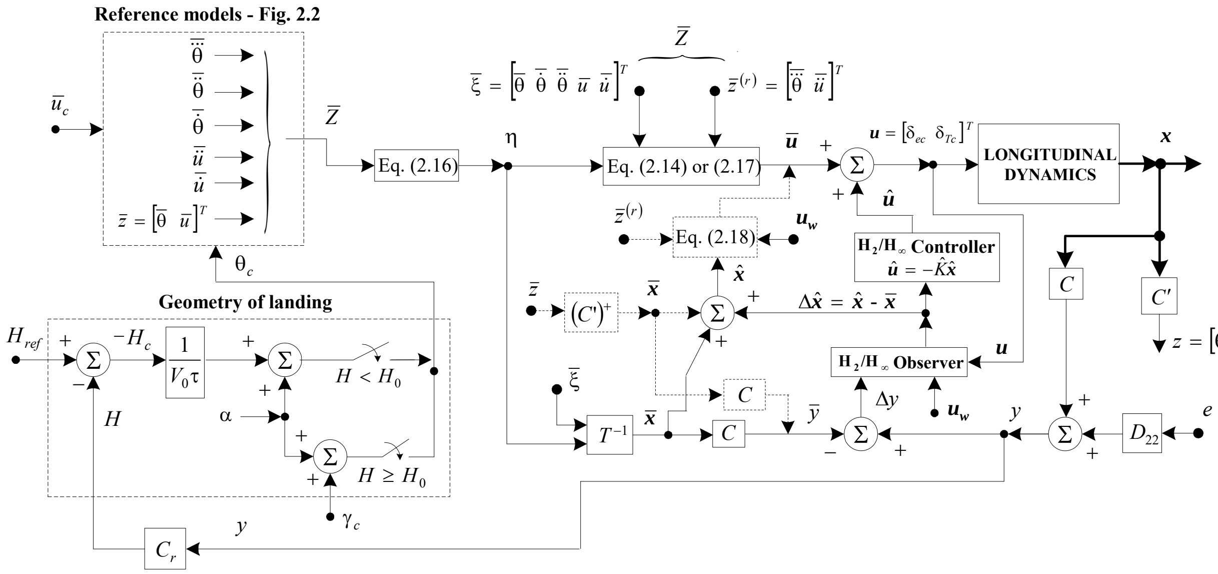


Fig. 2.6. Automatic landing subsystem for aircraft control in longitudinal plane using the H_2/H_∞ control and the dynamic inversion

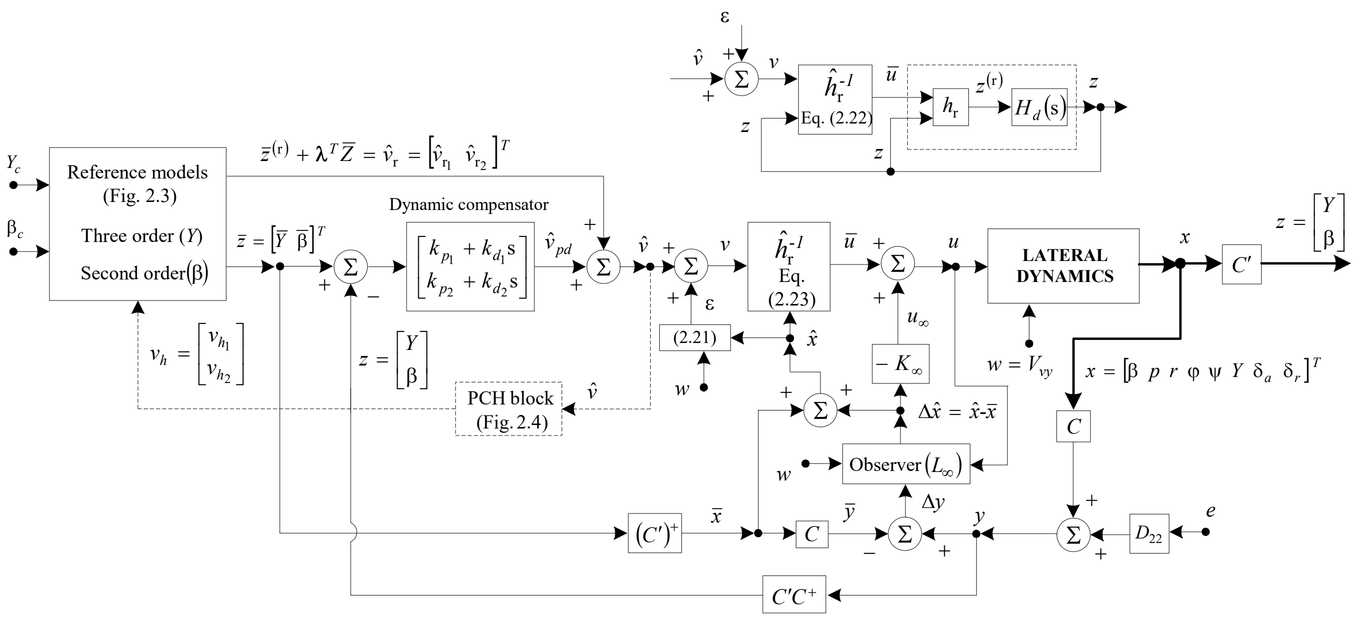


Fig. 2.7. Automatic landing subsystem for aircraft control in lateral-directional plane using the H-inf control and the dynamic inversion

optimal observer must assure the convergences: $\Delta y = y - \bar{y} \to 0 \left(y = C\boldsymbol{x} \to \bar{y} = C\bar{\boldsymbol{x}}, \boldsymbol{x} \to \bar{\boldsymbol{x}}\right), \Delta \boldsymbol{x} = \boldsymbol{x} - \bar{\boldsymbol{x}} \to 0 \ \left(\hat{\boldsymbol{x}} \to \boldsymbol{x} \to \bar{\boldsymbol{x}}\right)$; here, $\bar{\boldsymbol{x}}$ is aircraft desired state, while $\bar{y}$ is the reference vector associated to the measured output y.

The structure of the second control subsystem (lateral-directional plane) for aircraft guidance during the landing approach phase, using dynamic inversion, H-inf method, and a dynamic compensator is presented in Fig. 2.7.

Activity II.5. Interconections of the two optimal subsystems and the obtaining of a new system for the control of aircraft landing

For a Boeing 747 flight in longitudinal plane, the dynamics has been borrowed from [22]: $a_{11} = -0.021, a_{12} = 0.122, a_{14} = -0.322$, $a_{21} = -0.209, a_{22} = -0.53, a_{23} = 2.21, a_{24} = 0, a_{31} = 0.017, a_{32} = -0.164, a_{33} = -0.412, a_{34} = 0, b_{11} = 0.01, b_{12} = 1, b_{21} = -0.064, b_{22} = -0.044$, $b_{31} = -0.378$, $b_{32} = 0.544, u_c = \bar{u}_c = \bar{u} = V_0 = 70\,\text{m/s}, T_e = T_T = 0.9\,\text{s}$, $\tau = 4, c_1 = 0.31, c_2 = 3.16, \mu_1 = 50, \mu_2 = 100$. Because the H_2/H_∞ technique deals with linearized dynamics, use used here the linear dynamics of aircraft. The advantage of this technique is that it handles the plants having sensor errors and other disturbances – a real problem during aircraft landing. We neglected the time-delays and we considered the biases of the sensors – the only sensor errors (the measurement noises have been not taken into

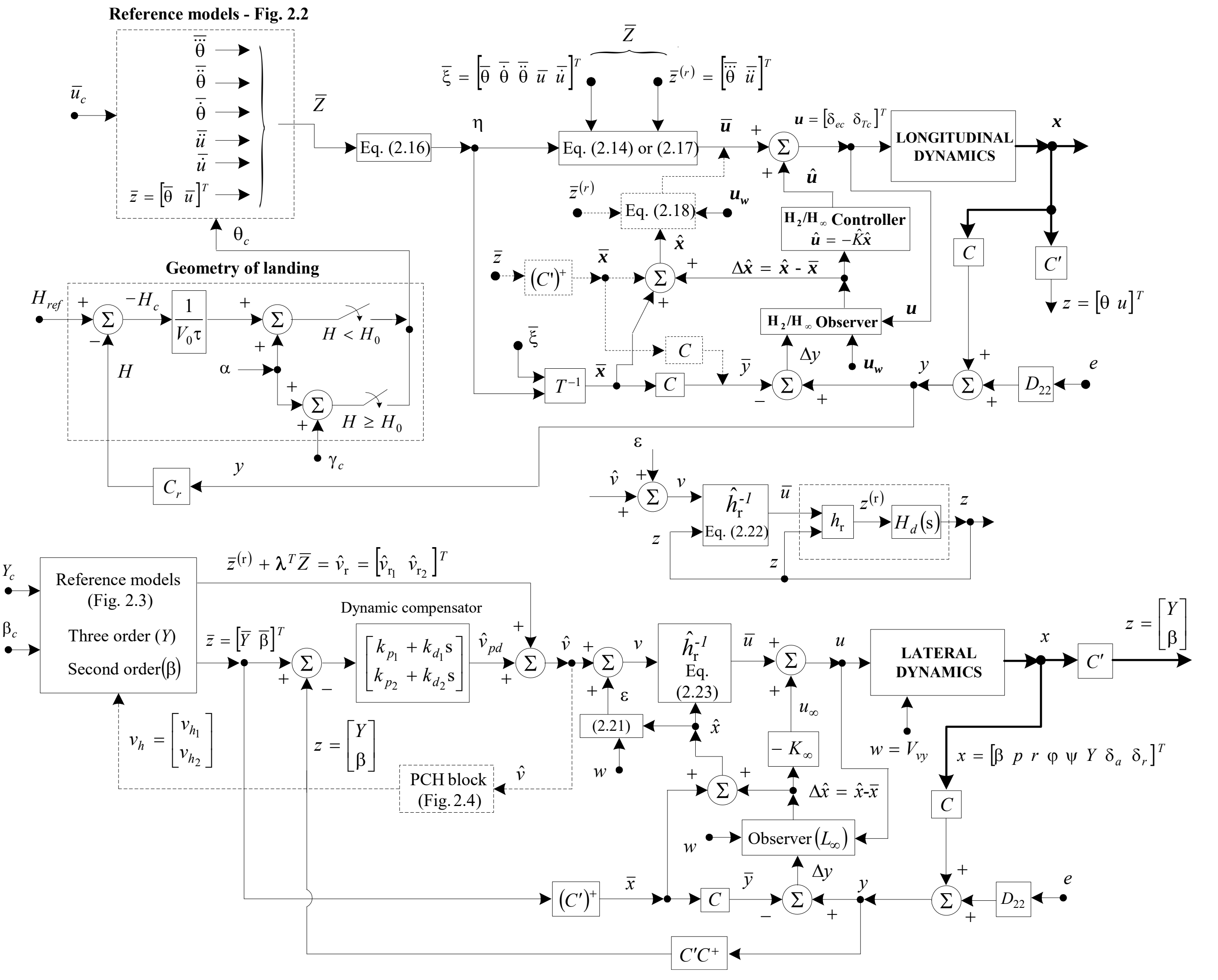


Fig. 2.8. Structure of the new architecture for aircraft automatic control during landing

account). The elements of matrix G have been calculated with (5); for the two stages of landing, the vector containing the sensor errors (biases associated to the measurement process) is

$$e = \begin{bmatrix} 0.1\text{m/s} & 0.1\text{deg} & 0\text{deg/s} & 0.1\text{deg} & 0\text{m} & 0.1\text{deg} & 0.1\text{deg} \end{bmatrix}^T,$$

while, for the reference models, one has chosen: $p = 25, \xi_1 = \xi_2 = 0.7, \omega_1 = \omega_2 = 2\,\text{rad/s}$. The values considered for the sensor errors are chosen very large because it is important to use strong disturbances instead of small ones when designing a robust ALS.

The following initial states were considered: $\boldsymbol{x}(0)=[1.0143 \quad -0.5\,\text{deg} \quad 2\,\text{deg/s} \quad -2.5\,\text{deg} \quad 6\,\text{s} \quad 0 \quad -3\,\text{deg} \quad 2\,\text{deg}]^T$ (glide slope) and $\boldsymbol{x}(0)=[1 \quad -0.1\,\text{deg} \quad 0\,\text{deg/s} \quad -2.5\,\text{deg} \quad 0.428\,\text{s} \quad 0 \quad -2\,\text{deg} \quad 0\,\text{deg}]^T$ (flare), espectively.

For lateral plane, we considered that the values of the coefficients in aircraft dynamics are [6]:

$$a_{11}=-0.0013, a_{12}=0, a_{13}=-1, a_{14}=0.15, a_{21}=-1.33, a_{22}=-0.98, a_{23}=0.33, a_{31}=0.17, a_{32}=-0.17, a_{33}=-0.217,$$
$$b_{11}=0.001, b_{12}=0.015, b_{21}=0.23, b_{22}=0.06, b_{31}=0.026, b_{32}=-0.15, V_0=67\,\text{m/s}, T_a=0.7\,\text{s}, T_r=0.1\,\text{s},$$
$$\mu_1=1, \mu_2=1, c_1=c_2=0.01, \overline{Y}=0\,\text{m}, \overline{\beta}=0\,\text{deg};$$

the vector e is

$$e=[0\,\text{m} \quad 0\,\text{m/s} \quad 1\,\text{deg} \quad 0\,\text{deg} \quad 1\,\text{deg/s} \quad 0\,\text{deg} \quad 1\,\text{deg/s}]^T,$$

the matrix G has been obtained by means of equation (3.25), while the system's initial state is

$$x(0)=[0.1\,\text{deg} \quad 0\,\text{deg}/s \quad -2\,\text{deg/s} \quad 0\,\text{deg} \quad 0.1\,\text{deg} \quad 25\,\text{m} \quad 0\,\text{deg} \quad 0\,\text{deg}]^T;$$

for the reference models, we have chosen: $\text{p}=25, \xi_1=\xi_2=0.7, \omega_1=\omega_2=2\,\text{rad/s}$.

The structure of the new automatic landing system, using dynamic inversion and H_2/H_∞ method, is presented in Fig. 2.8; it consists of two subsystems – the first one controls aircraft motion in longitudinal plane (the one in Fig. 2.6), while the second one is for the control of aircraft motion in lateral-directional plane (the one in Fig. 2.7).

Activity II.6. Software implementation of the system for the landing optimal control

To study the performances of the obtained automatic landing system, one considers the landing of a Boeing 747. Complex simulations in Matlab/Simulink environment have been performed; thus, one designed the optimal observers, the H_2/H_∞ controllers, the component obtained with the dynamic inversion method, and, after that, validated the proposed automatic landing system. The Matlab/Simulink model for the first phase of landing in longitudinal plane (glide slope) is presented in Fig. 2.9.a; it has two subsystems: 1) "Reference models + landing geometry" (Fig. 2.9.b); 2) "Aircraft dynamics" (Fig. 2.9.c).

The Matlab/Simulink model for the second phase of landing in longitudinal plane (flare) is presented in Fig. 2.10.a; it has two subsystems: 1) "Reference models + landing geometry" (Fig. 2.10.b); 2) "Aircraft dynamics" (Fig. 2.10.c).

The Matlab/Simulink model for the landing in lateral directional plane is presented in Fig. 2.11.a; it has four subsystems: 1) "Reference models" (Fig. 2.11.b); 2) "Aircraft dynamics" (Fig. 2.11.c); 3) "Observer and H-inf controller" (Fig. 2.11.d); 4) "Subsystem vh" (Fig. 2.11.e).

The Matlab program for validation of the complete landing system (longitudinal and lateral-directional planes) is presented in Appendix 1.

In figs. 2.12 and 2.13 we represent the time characteristics for the glide slope phase and flare phase, respectively. The last three mini-graphics in these two figures represent the deviations of the forward speed (u), slope angle (γ), and altitude (H), with respect to their nominal values, i.e. $\overline{u}-u, \gamma_c-\gamma, H_{ref}-H$. The time origin for the flare trajectory is chosen zero when the altitude is H=H_0=30 m (the altitude at which the glide slope phase ends).

From the theoretical part, we retained the mandatory values of the slope angle (the difference between the pitch angle and the attack angle): -2.5 degrees in the first landing phase and 0 degrees in the second phase, respectively. By analyzing Figs. 2.12 and 2.13, we remark the correctness of the simulation data. During the glide slope, the aircraft must have a linear descendent trajectory (fig. 2.12.g) and, as a consequence, the pitch angle must be negative; as one can see in Fig. 2.12, the pitch angle is -2.53 degrees, while the attack angle is slightly negative $(\cong -0.03\,\text{deg})$; it results the desired slope angle (-2.5 degrees). In the flare phase, the aircraft must describe a parabolic trajectory (Fig. 2.13.g) with a null slope angle; as one can see in Fig. 2.13, the pitch and the attack angles become zero in about 14 seconds; it results the desired null slope angle.

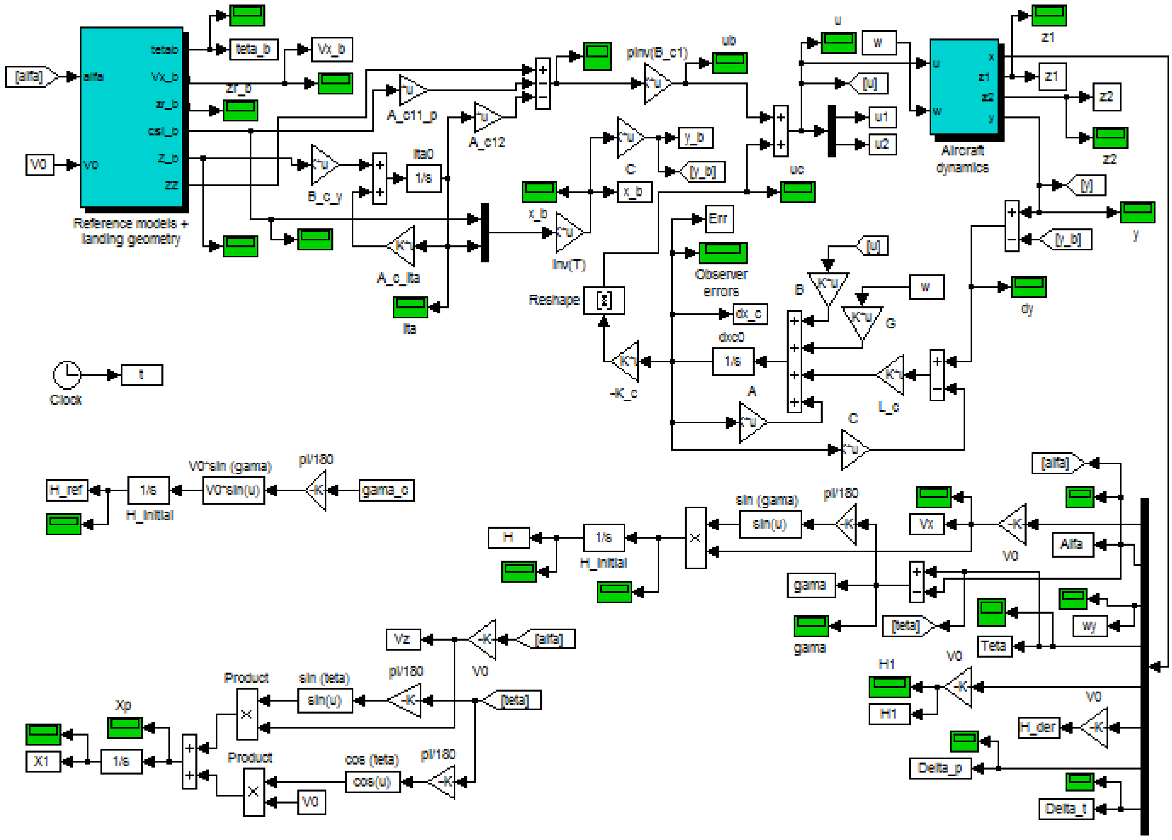


Fig. 2.9.a. Simulink model for glide slope phase (automatic landing subsystem for aircraft control in longitudinal plane)

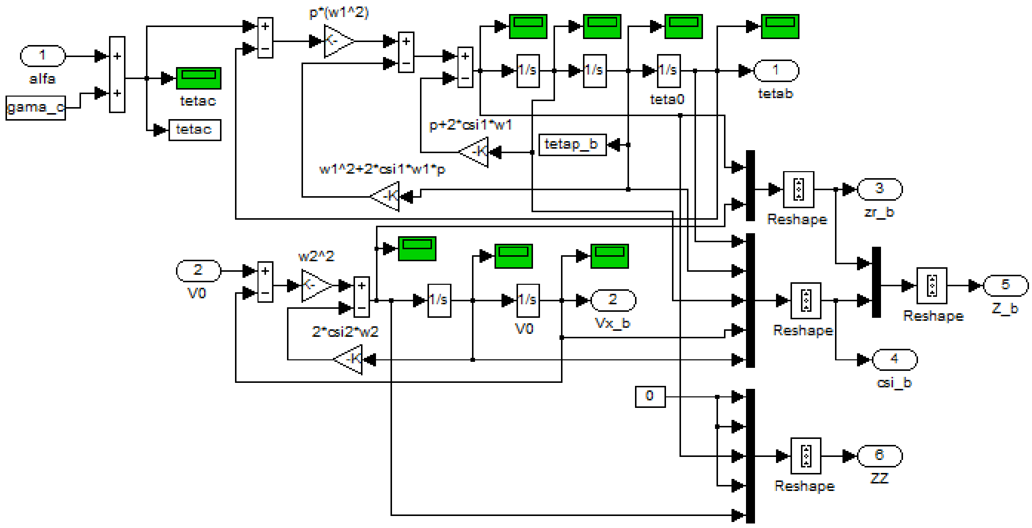


Fig. 2.9.b. “Reference models + landing geometry” Simulink model (glide slope phase)

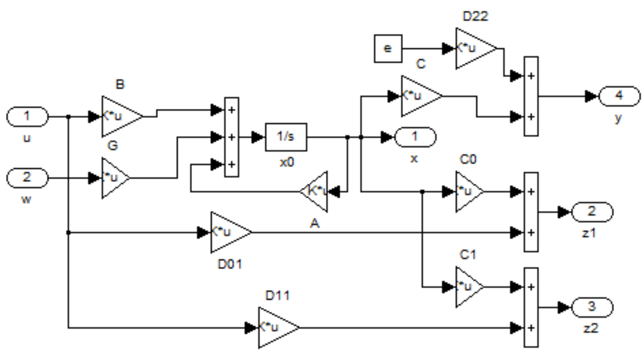


Fig. 2.9.c. “Aircraft dynamics” Simulink model (glide slope phase)

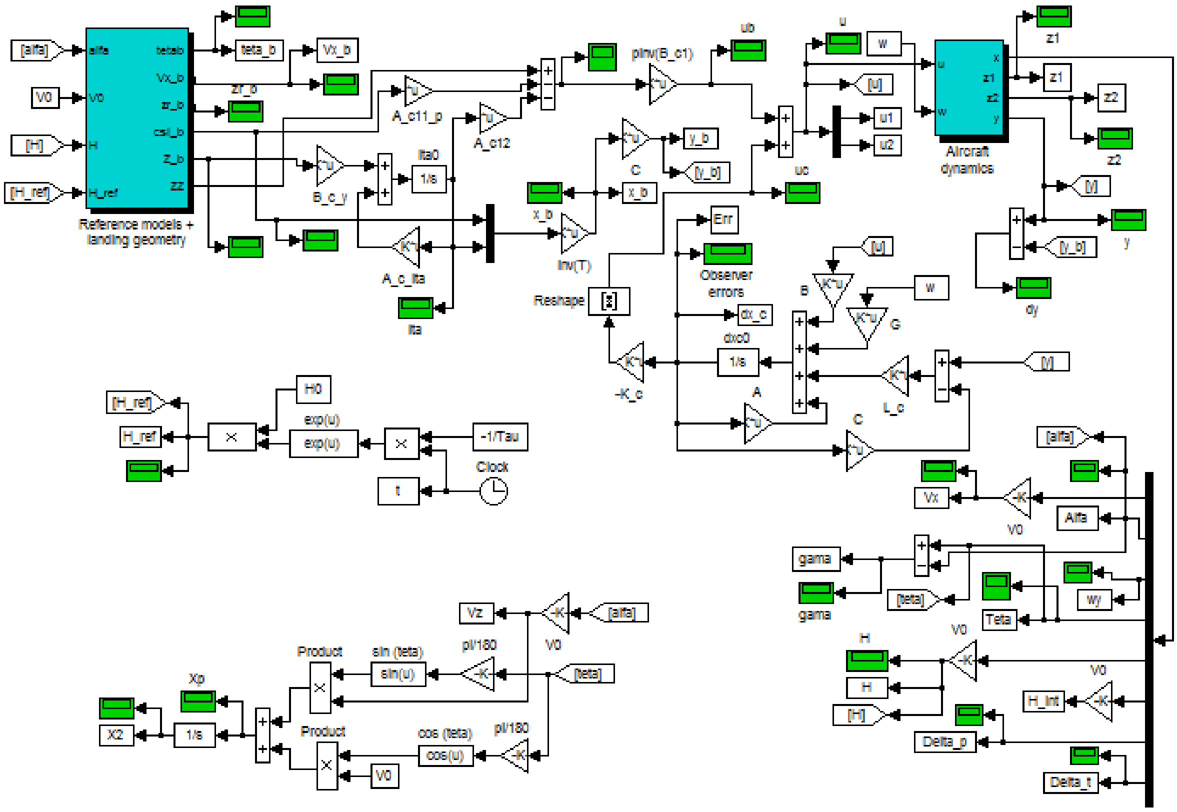


Fig. 2.10.a. Simulink model for flare phase (automatic landing subsystem for aircraft control in longitudinal plane)

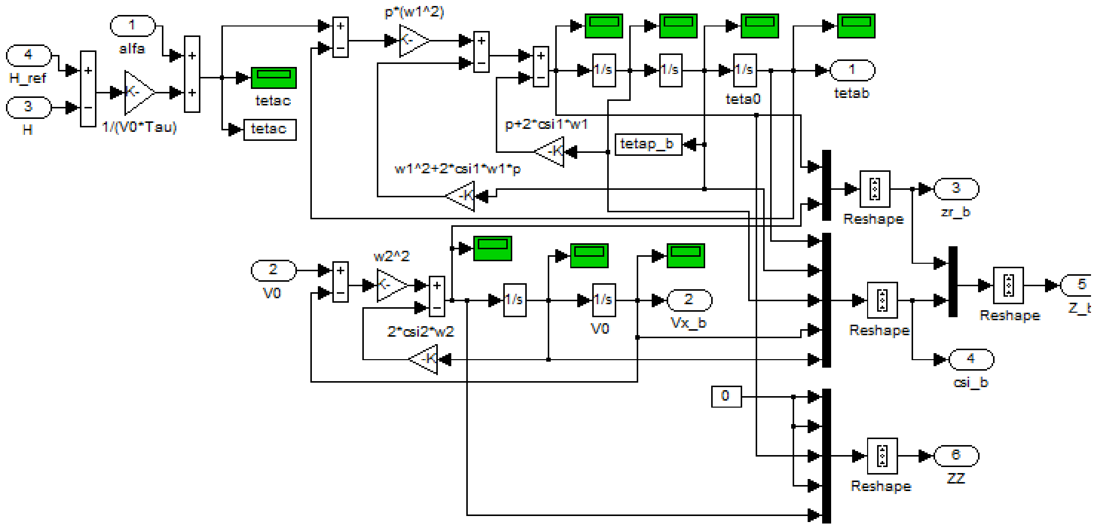


Fig. 2.10.b. “Reference models + landing geometry” Simulink model (flare phase)

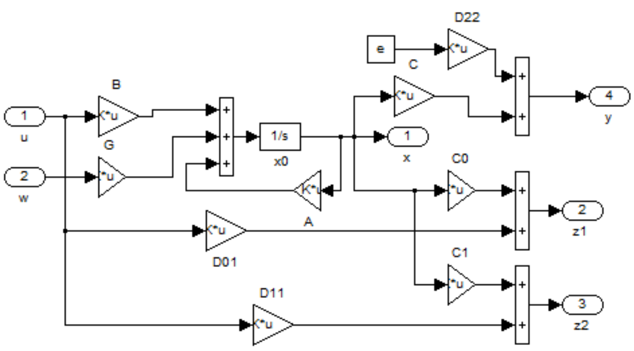


Fig. 2.10.c. “Aircraft dynamics” Simulink model (flare phase)

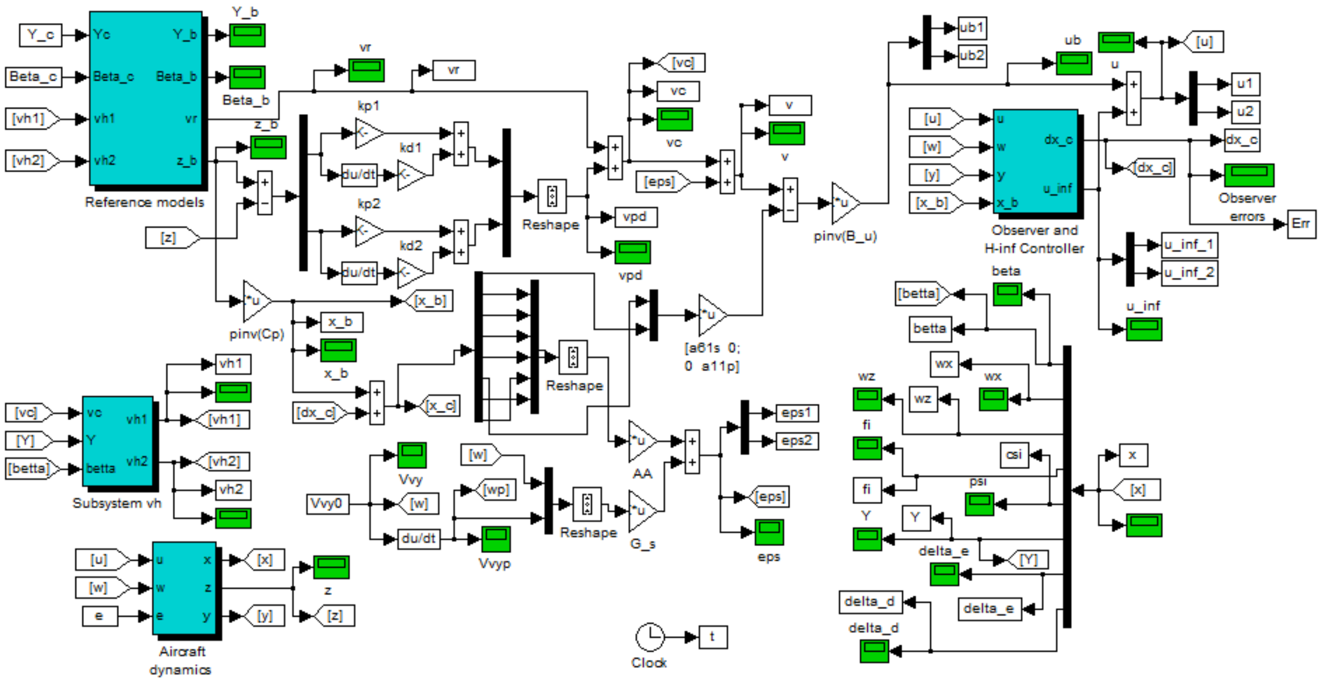


Fig. 2.11.a. Simulink model of the automatic landing subsystem for aircraft control in lateral-directional plane

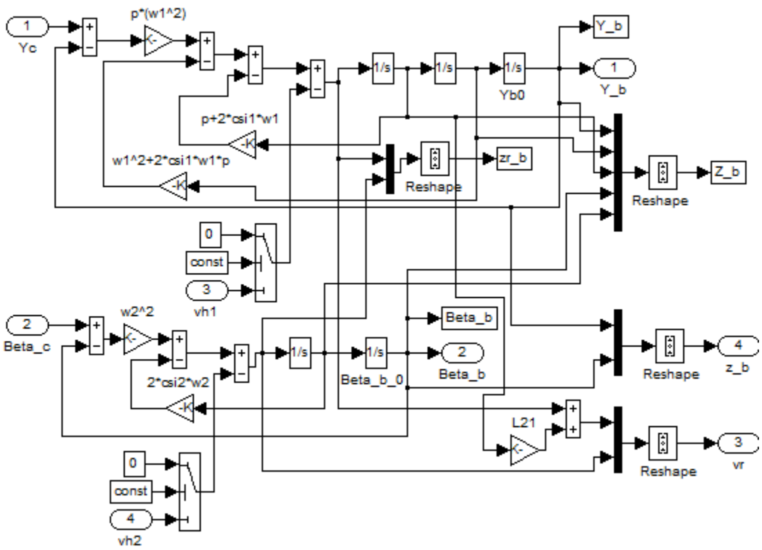


Fig. 2.11.b. Simulink model for “Reference models” (lateral-directional plane)

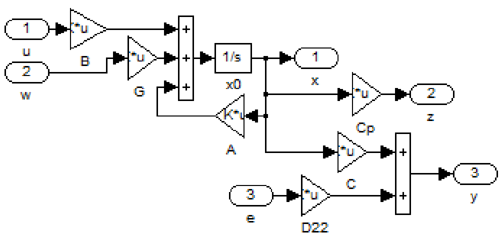


Fig. 2.11.c. Simulink model for “Aircraft dynamics” (lateral-directional plane)

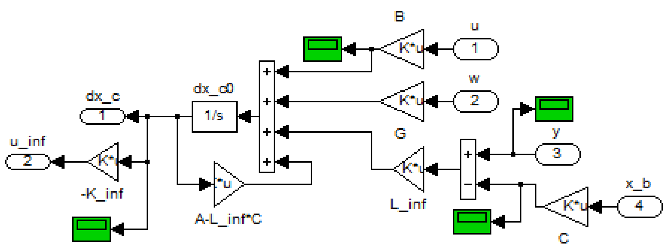


Fig. 2.11.d. Simulink model for “Observer and H-inf controller” (lateral-directional plane)

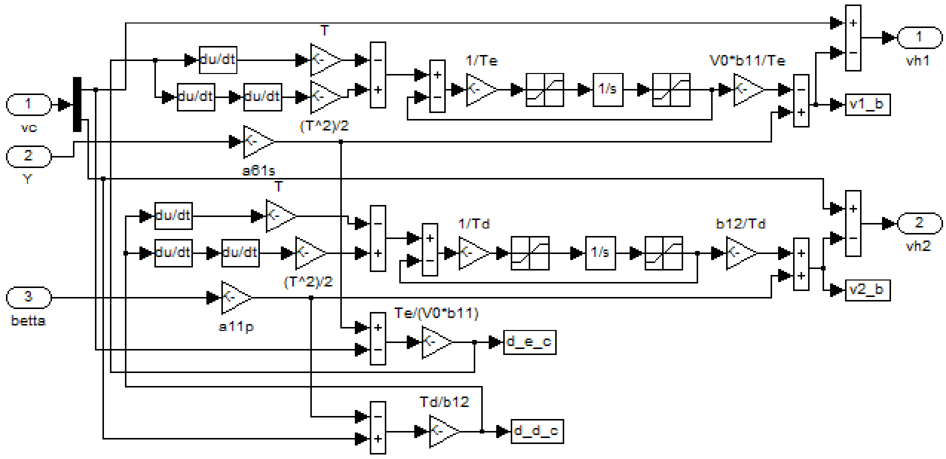


Fig. 2.11.e. Simulink model for " Subsystem vh" (lateral-directional plane)

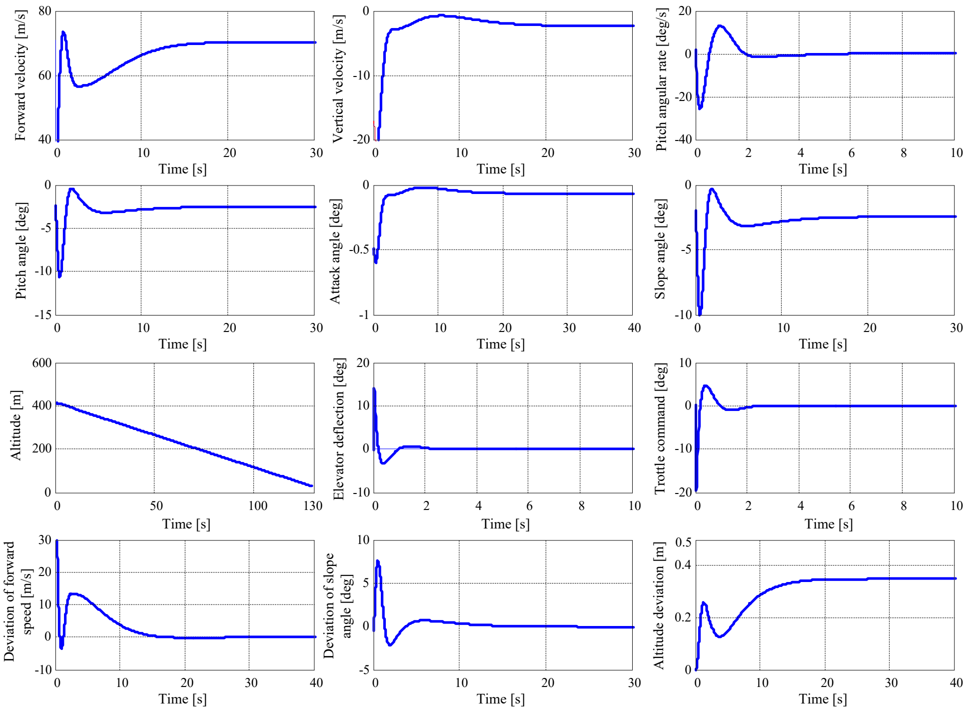


Fig. 2.12. Time characteristics for the glide slope phase (longitudinal plane)

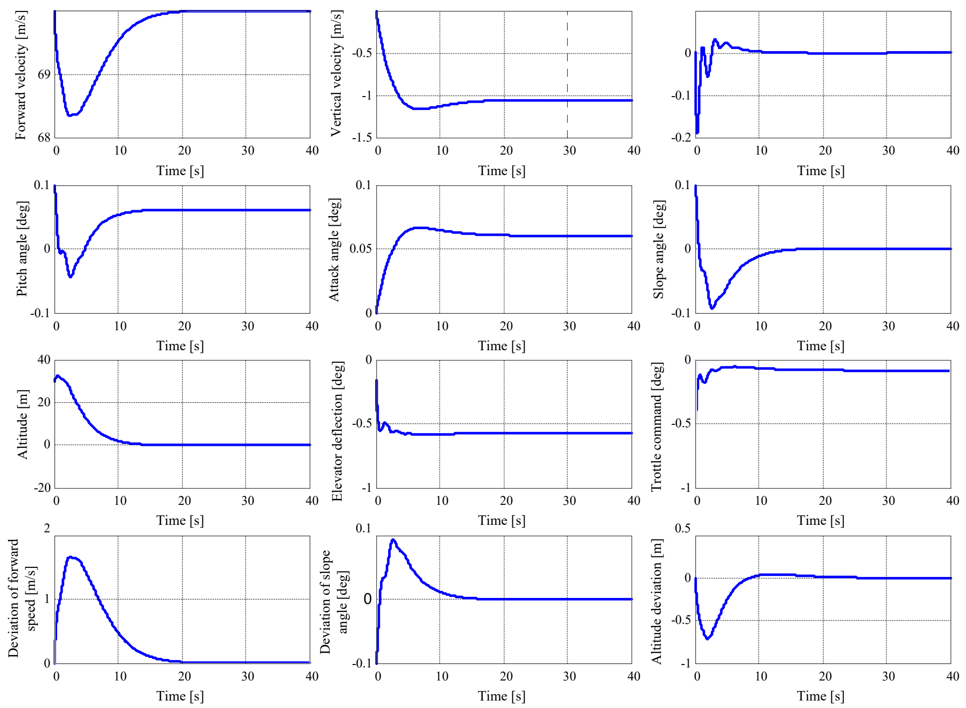


Fig. 2.13. Time characteristics for the flare phase (longitudinal plane)

The landing begins at a longitudinal speed initially exceeding the nominal speed by 1 m/s (see Fig. 2.12.a). The speed should be reduced to the normal speed (70 m/s) and then kept at this value; this landing process begins at 420 m (Fig. 2.12.g). From last graphic in Figs. 2.12 and 2.13, we can see that the final error between the desired path and the actual path is less than 0.35 m during the glide slope phase and very close to 0 m for flare. These errors are very small if the Federal Aviation Administration (FAA) accuracy requirements for Category III (the best category) [23] are analyzed; according to FAA Category III accuracy requirements, the vertical error (altitude deviation with respect to its nominal value) must be less than 0.5 m, the lateral deviation must be less than 4.1 m, while the final altitude at the end of flare must be 0 m. The robustness of our new automatic landing system is due to the H_2/H_∞ control technique, this method being capable to handle the plant with sensor errors and disturbances. From the forward speed point of view, the error is less than 0.01m/s (Fig. 2.12.j and fig. 2.13.j); because the deviation of the slope angle with respect to its nominal value $(\gamma_c - \gamma)$ tends to zero we conclude that $\gamma \to \gamma_c = -2.5\deg$, for the first landing phase, and $\gamma \to \gamma_c = 0\deg$ for the second landing phase. The optimal control system associated to aircraft flight during landing, based on H_2/H_∞ technique, assures the convergence $H \to H_{ref}$, for both cases when the sensor errors are taken or not into account. For Boeing 747, the first phase of the landing process takes approximately 129 seconds (Fig. 2.12.g), while the second phase of the landing (flare phase) takes approximately 25 seconds (Fig. 2.13.g); in the same time, the steady values of aircraft vertical velocity are $w \cong -2.2\,\text{m/s}$ (glide slope – fig. 2.3) and $w \cong -1.06\,\text{m/s}$ (flare – fig. 2.4), respectively. During the two stages of landing, the vertical velocity mean values are $w_{mean} \cong -3.02\,\text{m/s}$ and $w_{mean} \cong -1.2\,\text{m/s}$, respectively. Using this information, the vertical distance covered by the aircraft in the first landing phase must be approximately $3.02\text{m/s}\cdot 129\text{s} = 389.58\text{m}$, while the vertical distance covered in the second landing phase must be approximately $1.2\,\text{m/s}\cdot 25\,\text{s} = 30\,\text{m}$. These values are again confirmed by Figs. 2.12 and 2.13 since the glide slope phase means a 390 m descent for the aircraft center of gravity, while the flare phase means a 30 m descend.

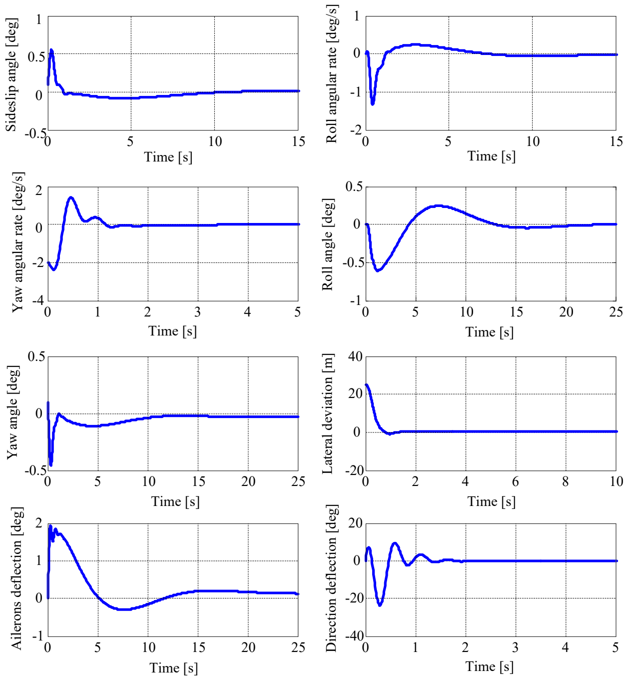


Fig. 2.14. Time characteristics of the lateral-directional auto landing control subsystem

For lateral-directional plane, by solving the two Riccati equations, it resulted P_∞ and P_∞^*; the calculation of the gain matrices K_∞ and L_∞ is made by means of (3.27) and $L_\infty = P_\infty^* C^T \left(D_{22}^T D_{22}\right)^{-1}$, respectively. The form of the command law $\bar{u}$ resulted from $\bar{u} = B_u^{-1}\left(\bar{z}^{(r)} - A_x \hat{x} - G' \tilde{w}\right)$; using the same Matlab program (Appendix 1), one obtained the observer estimation errors and the main variables' time history. In Fig. 2.14 we represent the time characteristics for the flight direction control subsystem (Fig. 2.7). The landing approach begins at the nominal speed (67 m/s); the speed should be maintained constant. To test the robustness of the first designed ALS, in all simulations, we have taken into consideration the crosswind, because low-altitude crosswind can be a serious threat to the safety of aircraft in landing. From 6th mini-graphic in Fig. 2.14 (achieved for V_{vy}=2 m/s), we can see that the stationary value of the aircraft lateral deviation (Y) is very close to zero; this error is very good if we analyze the Federal Aviation Administration (FAA) accuracy requirements for Category III (best category) [23]; according to FAA Category III accuracy requirements, the lateral error must be less than 4.1 m. If the lateral error is between 4.1 m and 4.6 m, the aircraft meets the Category II precision standards, while if the lateral error is between 4.6 m and 9.1 m, the aircraft meets Category I precision standards. The used technique (H-inf control) can handle the plant with measurement noise (sensor errors) and crosswind. If the crosswind is stronger than its maximum accepted value, the pilot must avoid having the aircraft enter into this wind shear. From the sideslip angle's point of view, the errors are less than 0.01 deg.; we conclude that $\beta \rightarrow \beta_c$=0 deg. The Matlab program for obtaining of Figs. 2.12-2.14 is presented in Appendix 1.

For longitudinal motion, the matrix D_{22} (the matrix of weights associated to the sensor errors) has been chosen as $D_{22} = \bar{k} I_7$ ($\bar{k} -$ positive constant, $\bar{k} = \det(D_{22})$); for Figs. 2.12 and 2.13, we have used the value $\bar{k} = 1$, but it is interesting to see what happens if this constant is increased. According to the expressions of the observer gain matrices L and L_∞, a change of the matrix D_{22} is equivalent with a modification of these two matrices and, according to the equation $\hat{L} = \hat{P}^* C^T \left(D_{22}^T D_{22}\right)^{-1}$, this change also means a modification of the observer final gain matrix $\left(\hat{L}\right)$, of the observer errors, and of all the variables' time history. In Figs. 2.15 and 2.16 we represent, beside the aircraft altitude, the time histories associated to the deviations of the altitude, forward speed, and slope angle, with respect to their nominal values, i.e. $H_{ref} - H, \bar{u} - u, \gamma_c - \gamma$ for different values of the matrix D_{22}; we have chosen 5 values for the constant $\bar{k}$, i.e.: 1, 1.1, 1.3, 1.4, and 1.6. The Matlab program for the obtaining of Figs. 2.15 and 2.16 is presented in Appendix 2.

For the first landing phase, as one can see in Fig. 2.15.a, the altitude decays faster for bigger values of the constant $\bar{k}$; this means that an increase of $\bar{k}$ is equivalent with the decrease of the glide slope duration and horizontal distance covered during this first stage of landing: the glide slope duration is 129 seconds for $\bar{k} = 1$, 128 seconds for $\bar{k} = 1.1$, 126.3 seconds for $\bar{k} = 1.3$, 125.5 seconds for $\bar{k} = 1.4$, and 124.4 seconds for $\bar{k} = 1.6$ (Fig. 2.15.a). This is because the aircraft speed increases together with the constant $\bar{k}$ (Fig. 2.15.c); a bigger speed, for a constant slope (Fig. 2.15.d), means a more rapid linear descend, but the negative effect is that the error $\bar{u} - u$ has increased from zero $\left(\bar{k} = 1\right)$ to -2.78 m/s $\left(\bar{k} = 1.6\right)$. Because H_{ref} is constant, the increase of the constant $\bar{k}$ is also equivalent with the increase of the aircraft altitude error (Fig. 2.15.b); this altitude error slightly increases

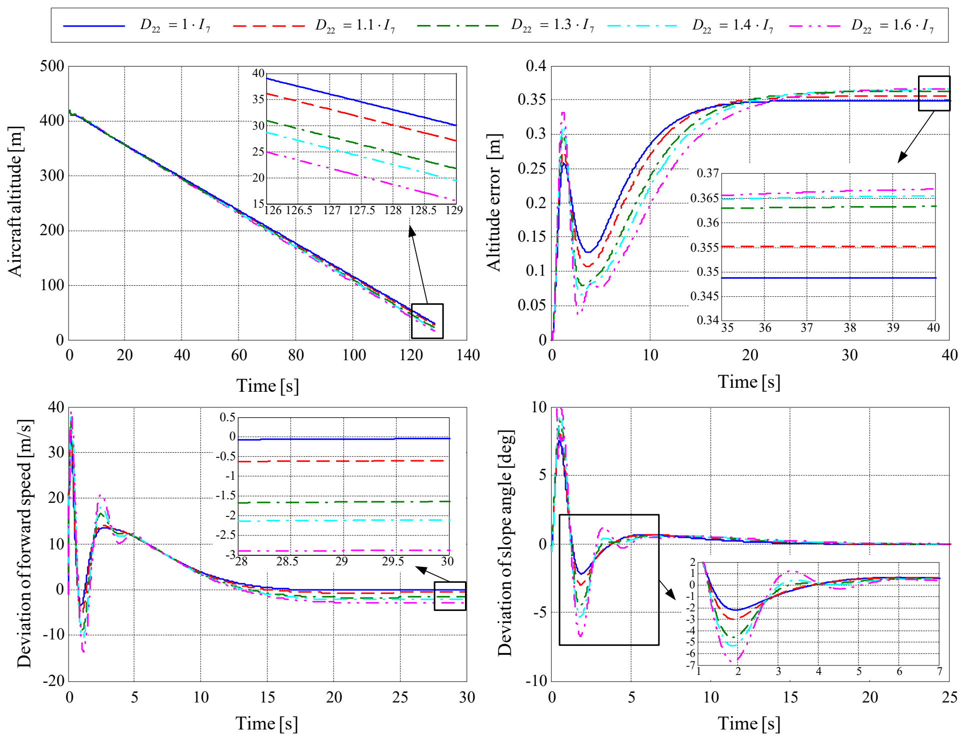


Fig. 2.15. Aircraft altitude and the main errors during glide slope for different matrix D_{22}

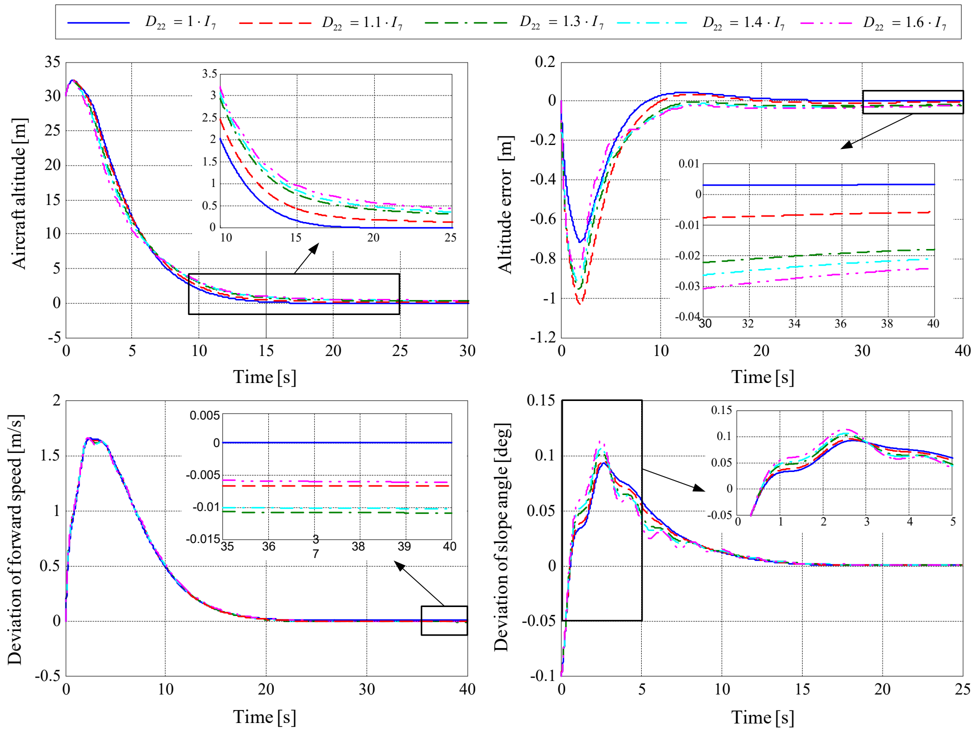


Fig. 2.16. Aircraft altitude and the main errors during flare for different matrix D_{22}

together with the constant $\bar{k}$ from 0.35 to 0.365. On the other hand, the variation of $\bar{k}$ does not modify the glide slope angle (Fig. 2.15.d), but it increases the overshoots and the transient regimes. For the flare stage, as one can see in the Fig. 2.16.a, the influence of D_{22}'s modification is different especially from the speed and flare duration point of view; thus, the increase of the constant $\bar{k}$ means: 1) the increase of the altitude stationary value from a value very close to 0 m to 0.25 m (Figs. 2.16.a and 2.16.b); 2) the increase of the altitude error absolute value $\left|H_{ref} - H\right|$ from a value very close to 0 m to 0.25 m (Fig. 2.16.b); 3) a small decrease of the aircraft speed, i.e. 0.01 m/s (Fig. 2.16.c), this explaining the increase of the flare duration (Fig. 2.16.a); 4) the increase of the overshoots and transient regimes especially for the glide slope angle (Fig. 2.16.d); 5) the flare duration and the horizontal distance covered during this stage of landing are bigger, while the appearing hump is bigger, this being an important disadvantage because, during the flight to small altitudes, the aircraft may be susceptible to accidents.

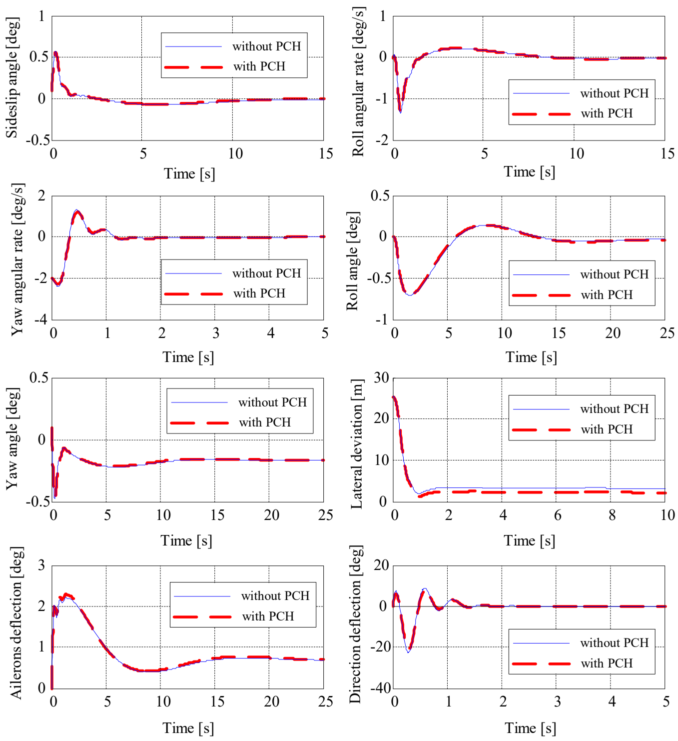


Fig. 2.17. Time characteristics of the lateral-directional control subsystem, with nonlinear actuators, with or without PCH

If the actuators are nonlinear, their models are the ones presented in Fig. 2.4; in this case, it is good to use a PCH block $(v_h \neq 0)$, because it allows the system to work in the linear zones of the nonlinearities. As we already presented above, the signal provided by the PCH block $v_h = \begin{bmatrix} v_{h_1} & v_{h_2} \end{bmatrix}^T$ is a reference model's additional input. In the case of nonlinear actuators, we analyze the influence of Pseudo Control Hedging usage on the second ALS's variables, by representing in Fig. 2.17 the time characteristics for the flight direction control subsystem with nonlinear actuators, affected by large crosswind (V_{vy}=10 m/s), in the presence of the sensors' errors, taking or not into account the signals from the PCH block.

Analyzing Fig. 2.17, we can remark that the usage of a PCH block improves the system's behavior; this is obvious especially in the case of the lateral deviation *Y*. In order to emphasize this, we plotted in Fig. 2.18 the time history of the lateral deviation with respect to the runway (*Y*), for different values of the lateral wind, using or not a PCH block when the actuators are nonlinear. Thus, from Fig. 2.18 we conclude that the differences between the dashed line curves (*Y* with PCH) and the solid line ones (*Y* without PCH) are easier to seen for large values of the crosswind; thus, the difference between the two curves is 0.25 m for V_{vy}=4 m/s, 0.3 m for V_{vy} =6 m/s, 0.4 m for V_{vy} =8 m/s, and 0.5 m for V_{vy} =10 m/s, respectively. This means that, in the case of nonlinear actuators' usage, the PCH block decreases the lateral deviation error, especially for large values of the crosswind, this improving the quality of the aircraft lateral-directional control system. The Matlab program for the obtaining of Figs. 2.17 and 2.18 is presented in Appendix 3.

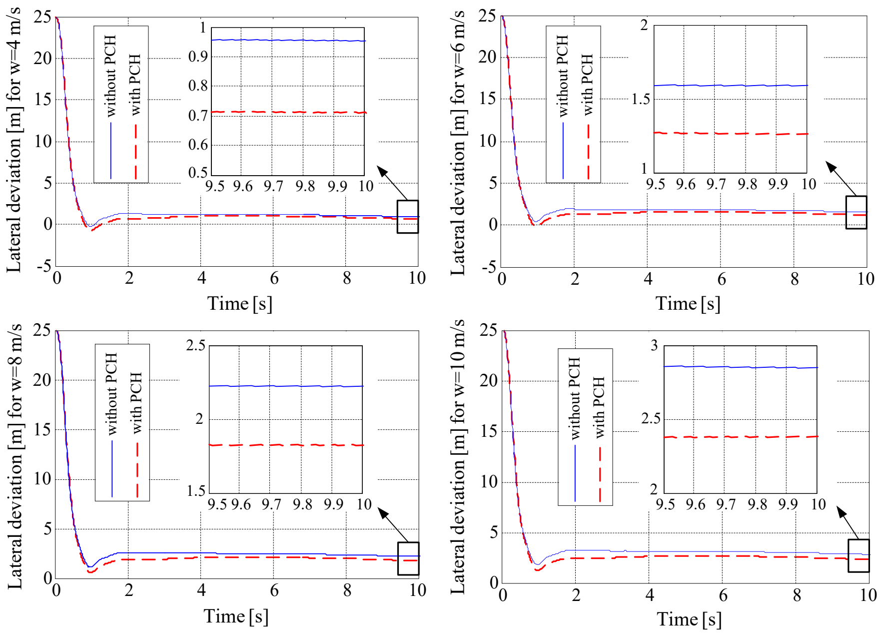


Fig. 2.18. Aircraft lateral deviation (*Y*), for different values of the lateral wind, using or not a PCH block

For longitudinal plane, in the expressions of $\hat{P}$ and $\hat{P}^*$ the constant *k* has been chosen 0.5, but it is also interesting to analyze the effect of this constant's modification on the altitude and altitude deviation with respect to its nominal value. Actually, the constant *k* represents the weight of the matrices P_∞ and P_∞^*, while the constant (1-*k*) represents the weight of the matrices *P* and P^* in the matrices $\hat{P}$ and $\hat{P}^*$ calculated by means of the H_2/H_∞ technique. With other words, the constant *k* may be regarded as the weight of the H_∞ technique in the control law design, while the constant (1-*k*) may be regarded as the weight of the H_2 technique in the control law design. If *k*=1, the control law is designed only by using the H_∞ technique, while the control law is designed only by using the H_2 technique if *k*=0. Till now, the weights of H_2 and H_∞ techniques have been considered equal (*k*=1-*k*=0.5); now, we increase and decrease the value of constant *k* and we analyze which is the effect of this modification on the aircraft altitude, altitude error, speed error, and glide slope error. These characteristics, in the presence of sensor errors, are represented in Fig. 2.19 (glide slope phase) and Fig. 2.20 (flare phase). Analyzing the characteristics' families in Fig. 2.19, we remark that the increase of the constant *k* (*k*=0.9) leads to the decrease of the aircraft speed with 0.6 m/s under its desired value (Fig. 2.19.c), while the glide slope angle is insignificantly affected (Fig. 2.19.d); in these circumstances, a lower speed means here the increase of the glide slope duration with 1 second (Fig. 2.19.a) and of the horizontal distance covered during this first stage of landing with 70 m. On the other hand, the decrease of constant *k* (*k*=0.1) leads to the increase of the aircraft speed with 0.2 m/s (Fig. 2.19.c), to an insignificant decrease of the glide slope duration with 0.32 seconds, and to the decrease of the horizontal distance covered during this first stage of landing with $70\,\mathrm{m/s} \cdot 0.32\,\mathrm{s} = 22.4\,\mathrm{m}$. Analyzing Fig. 2.19.b, one can conclude that the altitude error is not affected by the choice of constant k, while from the second and the fourth graphics in Fig. 2.19, one can notice that the increase of k means smaller overshoots although the differences are difficult to be visualized. The Matlab program for the obtaining of Figs. 2.19 and 2.29 is presented in Appendix 4.

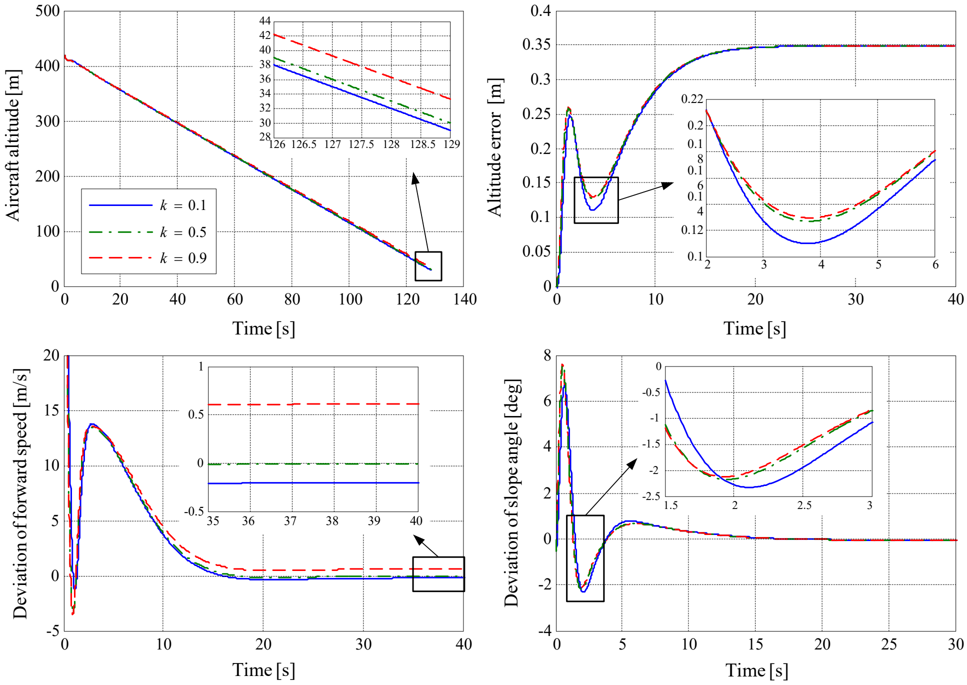


Fig. 2.19. Aircraft altitude and the main errors during glide slope for different values of the constant k

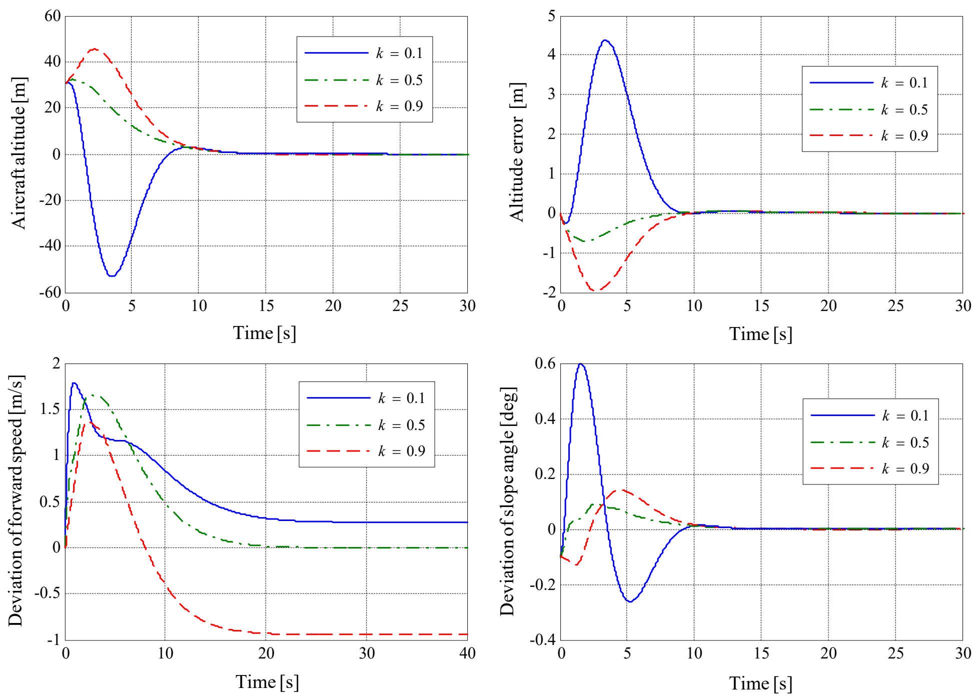


Fig. 2.20. Aircraft altitude and the main errors during flare for different values of the constant k

Analyzing now Fig. 2.20 (flare phase) we remark that the exponential curve for the altitude is obtained only for k=0.5 (the weights of H_2 and H_∞ techniques are equal in the control design); increasing or decreasing the constant k, we notice that the altitude tends to zero after an important hump (Fig. 2.20.a). During flare, the altitude must be described by null overshoot, this being achieved only for k=0.5. Moreover, the modification of k does not modify the final value of the altitude error but it dramatically increases the overshoot (Fig. 2.20.b). On the other hand, the flare duration, the horizontal distance covered during this stage of landing, and the glide slope angle are insignificantly influenced by the modification of the weight k, but the speed error is no more zero for k=0.1 or k=0.9; thus, for k=0.1, the speed error is 0.275 m/s, while, for k=0.9, this error is even bigger (-0.95 m/s). The conclusion is clear: the only value of k for which the both stages of landing are covered well is k=0.5.

Activity II.7. Organizing of a special session within IEEE International Conference - ICATE'16

The seventh activity during stage II of the project was the organizing of a special session for aircraft landing during a very good indexed international conference. This purpose has been achieved within the ***IEEE International Conference on Applied and Theoretical Electricity 2016 (ICATE 2016)*** – *ISI indexed conference*. Organized every two years starting with 1991, the International Conference on Applied and Theoretical Electricity has increased constantly in terms of organizational facilities, number of participants from Romania and abroad, scientific quality and visibility for academia and industry at regional and European level. The technical program committee was renewed over time with prominent specialists and the criteria of paper acceptance become more and more restrictive. As recognition of the increased scientific level, the two last editions (ICATE2012, ICATE2014) benefited by the IEEE technical co-sponsorship (IEEE Conference Record Numbers: #20803, #32678), and the conference proceedings were included in IEEEXplore database and Thomson Reuters ISI Conference Proceedings Citation Index. The web page of the conference is http://elth.ucv.ro/icate/icate16/.

The name of the special session organized within ICATE 2016 was „***Electrical Engineering in Transportation Systems (1) – Systems and equipment for aircraft landing***"; one of the session chairs was Prof. Romulus Lungu (member of our research team). A proof is presented below (Fig. 2.21 – list with regular sessions witin ICATE 2016; Fig. 2.22 – the list of papers accepeted for presentation in the special session).

The special session included 6 papers (see Fig. 2.22); two of them (**papers 10.2 and 10.6**) have the Acknowledgment of our project (grant no. 89/1.10.2015 - Modern architectures for the control of aircraft landing). In the following pictures, we present images from the session of papers' oral presentation.

ICATE 2016 – ***Conference program***

Location / Hours	University of Craiova, Faculty of Law Calea Bucuresti 107D, Craiova			
	Thursday, October 6			
9:00-10:00	Registration			
10:00-10:30	Opening ceremony / location Aula Magna			
10:30-11:00	Coffee Break / location Aula Magna lobby			
11:00-12:30	Plenary session – Keynote Speeches / location Aula Magna			
13:00-14:00	Lunch / location Emma Est Hotel, Calea Bucuresti 82A, Craiova			
	Workshops and Regular sessions			
	Room A	**Room B**	**Room C**	**Room D**
15:00-17:00	RS3. Control Systems Chairs: Teodor Pana Viorel Stoian	RS5. Electrical Machines and Drives (1) Chairs: Aurel Campeanu Mircea Radulescu	RS4. Power and Energy Systems (1) Chairs: Leszek S. Czarnecki Leonardo Geo Manescu	RS8. Industrial Applications (1) Chairs: Alexandru Morega Maria Brojbolu
17:00-17:30	*Coffee Break*			
17:30-19:30	WS3. New trends in electrical drive for intelligent traction systems Chairs: Petre Marian Nicoiae Sorin Enache	RS5. Electrical Machines and Drives (2) RS6. Power Electronics Chairs: Virgiliu Fireteanu Sergiu Ivanov	RS1. Circuits and Systems RS2. Magnetics Chairs: Dragos Niculae Elena Helerea	RS8. Industrial Applications (2) Chairs: Petru Notingher Virginia Ivanov
	Friday, October 7			
09:00-11:00	WS1. Safer EMF working environment in EU Chairs: Mihaela Morega Jolanta Karpowicz	WS2. Wireless Power Transfer Systems Chairs: Mihai Iordache Andrei Marinescu	RS4. Power and Energy Systems (2) Chairs: Florin Munteanu Stefan Gheorghe	RS10. Electrical Engineering in Transportation Systems (1) - Systems and equipment for aircraft landing Chairs: Romulus Lungu Stefan Luzica
11:00-11:30	*Coffee Break*			
11:30-13:30	WS1. Safer EMF working environment in EU RS7. Electromagnetic Compatibility and Engineering in Medicine and Biology Chairs: Calin Munteanu Lucian Mandache	RS8. Industrial Applications (3) RS9. Environment Engineering and Equipment Chairs: Constantin Bulac Dorin Lucache	RS4. Power and Energy Systems (3) Chairs: Mircea Chindris Denisa Rusinaru	RS10. Electrical Engineering in Transportation Systems (2) Chairs: Marius Minea Alexandru Tudosie
13:30-14:30	Lunch / location Emma Est Hotel, Calea Bucuresti 82A, Craiova			
15:00	Departure to Cetate – Culture Port for Gala Dinner. *Meeting point:* Emma Est Hotel, Calea Bucuresti 82A, Craiova			
17:00-23:00	Gala Dinner			
	Saturday, October 8			
10:00-12:00	Research facilities in Electrical Engineering – Guided tour			
12:00-12:30	Closing Ceremony, Aula Marius Preda, Faculty of Electrical Engineering			

Fig. 2.21. List with regular sessions witin ICATE 2016 (conference program)

ICATE 2016 – *Conference program*

RS 10	Electrical Engineering in Transportation Systems (1)
10.1	George Cristian Calugaru, Elena Andreea Danisor: *Dynamic Matrix Control Used in Stabilizing Aircraft Landing*
10.2	Mihai Lungu, Romulus Lungu, Lucian Grigorie, Octavian Preotu: *The Influence of Atmospheric Turbulences on Aircraft Landing Process*
10.3	Stefan Luzica, Radim Bloudicek: *The Hyperbolic Time Difference of Arrival Passive Surveillance System Analysis and Its Application for Precision Approach and Landing*
10.4	Alexandru Nicolae Tudosie: *Aircraft Engine with Coolant Injection into Its Compressor and Flow Rate Controller as Controlled Object*
10.5	Liviu Dinca, Jenica Ileana Corcau, Eduard Ureche: *Mathematical Modeling for Buck Converter in Continuous Conduction Mode*
10.6	Alexandru Nicolae Tudosie, Florentin Alin Butu: *Aircraft Landing With Decelerated Approach (Longitudinal Movement Model)*

Fig. 2.22. List of papers accepeted for presentation in our special session

Fig. 2.23. Images from the session of papers' presenting

Activity II.8. Results' dissemination

The 8th activity of project's second stage (2016) was intended to carry out the objective ***OS21*** (*To disseminate the results in the scientific, academic and socio-economic environment*). It was conducted throughout the stage and materialized through a web page actualization with all the information regarding the achievements during Stage II and the accomplishment of the project's objectives. The links for the web page are:

- http://elth.ucv.ro/site/mlungu/index.php?language=ro&page=48 (Romanian);
- http://elth.ucv.ro/site/mlungu/index.php?language=en&page=50 (English)

From the expected results' point of view, the research team had the following targets during 2016:

1) *Design of a new optimal control system using the dynamic inversion and H_2/H_∞ techniques*. **This aim of the project has been acomplished by designing the automatic landing system from Fig. 2.8**.
2) *Software package for the automatic landing system*. This aim of the project has been acomplished by obtaining the **Matlab/Simulink programs from Appendix (landing in two planes: longitudinal and lateral) and the Simulink models from Figs. 2.9, 2.10 and 2.11**.
3) *Phase repport* (**the present document**).
4) *Publishing of 2 papers in ISI Journals and 4 papers in other databases*. In 2016, the members of the research team have published **2 papers in ISI Journals, 5 papers in international databases (ISI Web of Science and IEEE Xplore), and 1 chapter in an international book (ISI Web of Science)**. Thus, **the target related to the papers' publishing has been reached and even exceeded**. Below, we present the the 7 papers (with a short description) published in 2016 by the members of the research team.

   4.1. Lungu, R., Lungu, M. *Design of Automatic Landing Systems using the H-inf Control and the Dynamic Inversion*. Journal of Dynamic Systems, Measurement and Control (Transactions of ASME), vol. 138, no. 2, 5 pp, 2016, ISSN: 0022-0434 (***ISI Journal***). *Databases: ISI Web of Science*.

   4.2. Lungu, R., Lungu, M. *Adaptive Flight Control Law Based on Neural Networks and Dynamic Inversion for Micro Aerial Vehicles*. Neurocomputing Journal, vol. 199, pp. 40-49, 2016, ISSN: 0925-2312 (***ISI Journal***). *Databases: ISI Web of Science*.

   4.3. Lungu, M., Lungu, R., Tutunea, D. *Control of Aircraft Landing using the Dynamic Inversion and the H-inf Control*. 17^{th} International Carpathian Control Conference (ICCC 2016), Tatranská Lomnica, Slovak Republic, May 29 - June 1, 2016, pp. 461-466. *Databases: IEEE Xplore*.

   4.4. Lungu, M., Lungu, R., Preotu, O. *Estimation of Aircraft State during Landing by means of Multiple Observers*. 23th International Conference on Systems, Signals and Image Processing (IWSSIP 2016), 23-25 May 2016, Bratislava, Slovakia. *Databases: IEEE Xplore*.

   4.5. Lungu, M., Lungu, R. *Reduced-Order Multiple Observer for Aircraft State Estimation during Landing*. 11^{th} edition of the International Conference on Trends in Aerospace, Robotics, Manufacturing Systems, Mechanical Engineering, Bioengineering, Power and Energy Engineering, Materials Engineering, Jupiter, 29 iunie - 2 iulie 2016; Applied Mechanics and Materials, vol. 841, 2014, pp. 253-259, 2016, DOI:10.4028/www.scientific.net/AMM.841.253 ISSN: 1660-9336 (***ISI Proceedings***). *In the indexing process in the databases: ISI Web of Science*.

   4.6. Lungu, M., Lungu, R., Grigorie, L., Preotu, O. *The Influence of Atmospheric Turbulences on Aircraft Landing Process*. International Conference on Applied and Theoretical Electricity – ICATE 2016 (***ISI Proceedings***). *In the indexing process in the databases: ISI Web of Science, IEEE Xplore*.

   4.7. Tudosie, A., Butu, A. *Aircraft Landing With Decelerated Approach (Longitudinal Movement Model)*. International Conference on Applied and Theoretical Electricity – ICATE 2016 (***ISI Proceedings***). *In the indexing process in the databases: ISI Web of Science, IEEE Xplore*.

   4.8. Lungu, R., Lungu, M. *Aircraft Landing Control Using the H-inf Control and the Dynamic Inversion Technique*. Chapter in the book „Automation and Control Trends", ISBN 978-953-51-2671-3 (editors: Pedro Ponce, Arturo Molina Gutierrez, Luis M. Ibarra). Intech Publisher, 2016, pp. 101-120. *In the indexing process in the databases: ISI Web of Science*.

Short description of the published papers:

- Lungu, R., Lungu, M. *Design of Automatic Landing Systems using the H-inf Control and the Dynamic Inversion*. Journal of Dynamic Systems, Measurement and Control (Transactions of ASME), vol. 138, no. 2, 5 pp, 2016, ISSN: 0022-0434 (***ISI Journal***). *Databases: ISI Web of Science*.

The paper focuses on the automatic control of aircraft in the longitudinal plane, during landing, by using the linearized dynamics of aircraft, taking into consideration the wind shears and the errors of the sensors. A new robust automatic landing system is obtained by means of the H-inf control, the dynamic inversion, an optimal observer and two reference models providing the aircraft desired velocity and altitude. The theoretical results are validated by numerical simulations for a Boeing 747 landing; the simulation results are very good (Federal Aviation Administration accuracy requirements for Category III are met) and show the robustness of the system even in the presence of wind shears and sensor errors. Moreover, the designed control law has the ability to reject the sensor measurement noises and wind shears with low intensity.

- Lungu, R., Lungu, M. *Adaptive Flight Control Law Based on Neural Networks and Dynamic Inversion for Micro Aerial Vehicles*. Neurocomputing Journal, vol. 199, pp. 40-49, 2016, ISSN: 0925-2312 (***ISI Journal***). *Databases: ISI Web of Science*.

The paper presents two new adaptive systems, for the attitude‘s control of the micro aerial vehicles (MAVs) – insect type. The dynamic model describing the motion of MAVs with respect to the Earth tied frame is nonlinear and the design of the new adaptive control system is based on the dynamic inversion technique. The inversion error is calculated with respect to the control law and two matrices (inertia and dynamic damping matrices) which express the deviation of the estimated matrices relative to the calculated ones (the matrices from the nonlinear dynamics of MAVs) in conditions of absolute stability in closed loop system by using the Lyapunov theory. To completely compensate this error, an adaptive component (output of a neural network) is added in the control law. The system also includes a second order reference model which provides the desired attitude vector and its derivative. The two variants of the new adaptive control system are validated by complex numerical simulations.

- Lungu, M., Lungu, R., Tutunea, D. *Control of Aircraft Landing using the Dynamic Inversion and the H-inf Control*. 17th International Carpathian Control Conference (ICCC 2016), Tatranská Lomnica, Slovak Republic, May 29 - June 1, 2016, pp. 461-466. *Databases: IEEE Xplore*.

The paper focuses on the automatic control of aircraft in the longitudinal plane, during landing, by using the literalized dynamics of aircraft, taking into consideration the wind shears and the errors of the sensors. The H-inf control provides robust stability with respect to the uncertainties caused by different disturbances and noise type signals, while the dynamic inversion provides good precision tracking. A new robust automatic landing system (ALS) is obtained by means of the H-inf control, the dynamic inversion, optimal observers, a dynamic compensator and two reference models providing the aircraft desired velocity and altitude. The theoretical results are validated using numerical simulations for a light aircraft landing; the simulation results are very good (Federal Aviation Administration accuracy requirements for Category III are met) and show the robustness of the algorithm even in the presence of wind shears and sensor errors.

- Lungu, M., Lungu, R., Preotu, O. *Estimation of Aircraft State during Landing by means of Multiple Observers*. 23th International Conference on Systems, Signals and Image Processing (IWSSIP 2016), 23-25 May 2016, Bratislava, Slovakia. *Databases: IEEE Xplore*.

The paper presents three multiple observers which are useful for the state reconstruction in the case of nonlinear systems characterized by unknown inputs. In the observers’ design process, the Lyapunov and linear matriceal inequality theories are used. The paper inovation is related to the use of Takagi-Sugeno multiple-model and multiple observers to the estimation of an aircraft state during the landing process. The validation of the three observers’ design algorithms is achieved through complex numerical simulations in Matlab/Simulink; the states and the unknown input vectors of the Takagi-Sugeno multiple-models are estimated and it is proved the proper functioning of the multiple observers as well as the very good estimation of the system’s states.

- Lungu, M., Lungu, R. *Reduced-Order Multiple Observer for Aircraft State Estimation during Landing*. 11th edition of the International Conference on Trends in Aerospace, Robotics, Manufacturing Systems, Mechanical Engineering, Bioengineering, Power and Energy Engineering, Materials Engineering, Jupiter, 29 iunie - 2 iulie 2016; Applied Mechanics and Materials, vol. 841, 2014, pp. 253-259, 2016, DOI:10.4028/www.scientific.net/AMM.841.253 ISSN: 1660-9336 (***ISI Proceedings***). *In the indexing process in the databases: Google Scholar, ISI Web of Science*.

The paper presents a new reduced-order multiple observer which can achieve the finite-time reconstruction of the system's state associated to a multiple-model. This observer is a combination of a reduced-order observer and a full-order multiple observer. The design of the new observer involves the usage of the Lyapunov theory, the solving of a linear matriceal inequality, and a variables' change. The steps of the design procedure have been software implemented in order to validate the new reduced-order multiple observer for the case of an aircraft motion during landing.

- Lungu, M., Lungu, R., Grigorie, L., Preotu, O. *The Influence of Atmospheric Turbulences on Aircraft Landing Process*. International Conference on Applied and Theoretical Electricity – ICATE 2016 (***ISI Proceedings***). *In the indexing process in the databases: ISI Web of Science, IEEE Xplore*.

The paper presents the automatic control of aircraft in the longitudinal (vertical) plane during landing, taking into account the atmospheric turbulences. The presented structure of automatic landing system consists of two coupled automatic control systems: the former is used for the control of aircraft longitudinal velocity, while the latter performs the control of the flight altitude by means of a subsystem for the pitch angle control; these 2 subsystems are the parts of the same auto-pilot. The control laws are based on the dynamic inversion concept with proportional- integral-derivative controllers in conventional and fuzzy variants. The theoretical results are validated by numerical simulations in the absence or in the presence of atmospheric turbulences.

- Tudosie, A., Butu, A. *Aircraft Landing With Decelerated Approach (Longitudinal Movement Model)*. International Conference on Applied and Theoretical Electricity – ICATE 2016 (***ISI Proceedings***). *In the indexing process in the databases: ISI Web of Science, IEEE Xplore*.

The paper deals with a mathematical model for airplanes' longitudinal movement during the approach-stage of the landing. One has described an aerodynamic decelerated approach and has established the correlation between airplane's flight commands and thrust level (engines' commands). Airplane's command law was issued (a common law, for rudder and throttle positions), according to the approach law, which should be implemented into its board computer for the landing phase of the flight; system's mathematical model was used for some simulations concerning its quality.

- Lungu, R., Lungu, M. *Aircraft Landing Control Using the H-inf Control and the Dynamic Inversion Technique*. Chapter in the book „Automation and Control Trends", ISBN 978-953-51-2671-3 (editors: Pedro Ponce, Arturo Molina Gutierrez, Luis M. Ibarra). Intech Publisher, 2016, pp. 101-120. *In the indexing process in the databases: ISI Web of Science*.

The chapter presents the automatic control of aircraft during landing, taking into account the sensor errors and the wind shears. Both planes – longitudinal and lateral-directional are treated; the new obtained automatic landing system (ALS) will consists of two subsystems – the first one controls aircraft motion in longitudinal plane, while the second one is for the control of aircraft motion in lateral-directional plane. These two systems can be treated separately but, in the same time, these can be put together to control all the parameters which interfere in the dynamics of aircraft landing. The two new automatic landing systems are designed by using the H-inf control, the dynamic inversion, optimal observers, and reference models. To validate the new obtained automatic landing system, one uses the dynamics associated to the landing of a Boeing 747, software implements the theoretical results and analyzes the accuracy of the results and the precision standards' achievement with respect to the requirements of the Federal Aviation Administration (FAA).

Also, in the last part of year 2016, other **four papers** have been sent for review and possible publish. ***Three of them have been sent to ISI Journals and one paper to an important international conference***. Below, we present their names and a short description of the papers.

- Lungu, R., Lungu, M. *Automatic Landing System using Neural Networks and Radio-technical Subsystems*. Chinese Journal of Aeronautics, ISSN: 1000-9361 (***ISI Journal***). Factor de impact relativ revista: 1.070.
- Lungu, M., Lungu, R. *Landing Auto-pilots for Aircraft Motion in Longitudinal Plane using Adaptive Control Laws Based on Neural Networks and Dynamic Inversion*. Asian Journal of Control, ISSN: 1561-8625 (***ISI Journal***). Journal relative impact factor:1.407.
- Lungu, R., Lungu, M. *Automatic control of the micro aerial vehicles' attitude and position*. International Journal of Micro Aerial Vehicles, ISSN: 1756-8293 (***ISI Journal***). Journal relative impact factor: 0.343.
- Lungu, M., Lungu, R. *Automatic Control of Aircraft Landing by using the H_2/H_∞ Control Technique*. The 36th IASTED International Conference on Modelling, Identification and Control (MIC 2017), February 20-22, 2017, Innsbruck, Austria.

Also it worth mentioning that some of the information obtained from this phase of the project were used to improve the Master course ***„Aircraft automatic flight control during landing"***, University of Craiova.

Taking into account all the issues presed above, for Phase II of the project (*Design, validation and optimizing of the optimal Auto Landing System*), we consider that all the eight specific activities were fully acomplished and all the 5 specific objectives of Phase II were achieved. Thus, the achievement of all the objectives originally set for this phase creates the premises to solve successfully the next stage of the project – *Design, validation and optimization of the adaptive ALS. Comparative studies between the two designed ALSs.* The first activity within Phase III - 2017 (*Acivity III.1. Optimization of the optimal ALS and robustness' improvement. Study of the sensor errors and atmospheric disturbances' influence*) will be a continuation of the work done in 2016; with other words, for the complete ALS designed in 2016, we will study the efect of sensor errors and atmospheric disturbances upon aircraft motion during landing and we will improve the robustness of the optimal automatic landing system with respect to these disturbances.

References (phase II of the project)

[1] Parkinson, B., O'Connor, M., Fitzgibbon, K. *Aircraft automatic approach and landing using GPS*, Global Positioning System: Theory and Applications, vol. II, pp. 397-425, 1996.

[2] Lungu, R., Lungu, M. *Automatic Landing Control using H-inf Control and Dynamic Inversion*, Proceedings of the Institution of Mechanical Engineers Part G - Journal of Aerospace Engineering, vol. 228, no. 14, 2014.

[3] Lungu, R., Lungu, M., Grigorie, T.L. *ALSs with conventional and fuzzy controllers considering wind shears and gyro errors*. Journal of Aerospace Engineering, vol. 26, no. 4, 2012, pp. 794-813.

[4] Lungu, R., Lungu, M., and Grigorie, T.L. *Automatic Control of Aircraft in Longitudinal Plane during Landing*. IEEE Transaction on Aerospace and Electronic Systems, vol. 49, no. 2, 2013, pp. 1338-1350.

[5] Lungu, M. *Sisteme de conducere a zborului (Flight control systems)*. Sitech Publisher, Craiova, 2008.

[6] Sadati, H., Sabzeh, I., Parvar, M., Menhaj, M.B. *Comparison of Flight control Systems Design Methods in Landing*, Asian Journal of Control, vol. 9, no. 4, 2007, pp. 491-496.

[7] Lungu, R., Lungu, M. *Automatic Control of Aircraft in Lateral-Directional Plane During Landing*. Asian Journal of Control, vol. 18, 2016, pp. 1-16.

[8] Lungu, M., Lungu, R. *Automatic Control of Aircraft Lateral-directional Motion during Landing using Neural Networks and Radio-technical Subsystems*. Neurocomputing Journal, vol. 171, 2016, pp. 471-481.

[9] Singh, S., Padhi, R. *Automatic Path Planning and Control Design for Autonomous Landing of UAVs using Dynamic Inversion*, American Control Conference Riverfront, St. Louis, MO, USA, 2009, pp. 2409-2414.

[10] Wagner, T., Valasek, J. *Digital Autoland Control Laws Using Quantitative Feedback Theory and Direct Digital Design*, Journal of Guidance, Control, and Dynamics, vol. 30, 2017, pp. 1399-1413.

[11] Kumar, V., Rana, K.P., Gupta, V. *Real-Time Performance Evaluation of a Fuzzy PI + Fuzzy PD Controller for Liquid-Level*

Process. International Journal of Intelligent Control and Systems, vol. 13, 2008, pp. 89-96.

[12] Lau, K., Lopez, R., Onate E. *Neural Networks for Optimal Control of Aircraft Landing Systems*. Proceedings of the World Congress on Engineering, vol. II, 2007, pp. 904-911.

[13] Zhi, L., Yong, W. *Intelligent landing of unmanned aerial vehicle using hierarchical fuzzy control*. IEEE Aerospace Conference, 2012, pp. 1-12.

[14] Ismail, S., Pashilkar, A., Ayyagari, R., Sundararajan, N. *Improved autolanding controller for aircraft encountering unknown actuator failures*. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2013, pp. 96-103.

[15] Shue, S., Agarwal, R.K. *Design of automatic landing systems using mixed H_2/H_∞ control*. Journal of Guidance, Control, and Dynamics, vol. 22, pp. 103-114, 1999.

[16] Stoica, A.M. *Disturbance Attenuation and its Applications*. Romanian Academy Publisher, 2004.

[17] Che, J., Chen, D. *Automatic Landing Control using H-inf control and Stable Inversion*. Proceedings of the 40th Conference on Decision and Control, Orlando, Florida, USA, 2001, pp. 241-246.

[18] Calise, A.J., Hovakymyan, N., Idan, M. *Output Control of Nonlinear Systems Using Neural Networks*, Automatica, vol. 37, no. 8, 2001, pp. 1201-1211.

[19] Isidori, A. *Nonlinear Control Systems,* Springer Publisher, Berlin, 1995.

[20] Lungu, R., Lungu, M., Rotaru, C. *Non-linear adaptive system for the control of the helicopters pitch's angle*. Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science, vol. 12, no. 2, 2011, pp. 133-142.

[21] Calise, A.J., Johnson, E.N., Johnson, M.D., Corban, J.E. *Applications of Adaptive Neural – Networks Control to Unmanned Aerial Vehicles*, Journal of Harbin Institute of Technology, vol. 38, no. 11, 2006, pp. 1865-1869.

[22] Yu, G.R. *Nonlinear Fly-By-Throttle H-inf Control using Neural Networks*, Asian Journal of Control, vol. 3, no. 2, 2001, pp. 163-169.

[23] Braff, R., Powell, J.D., Dorfler, J. *Applications of GPS to air traffic control*. Global Positioning System: Theory and Applications, vol. II, 1996, pp. 327-374.

SYSNTHESYS OF THE WORK,

containing the activities and the obtained results in comparison with the objectives of the project

for

- Stage III (2017) –

Stage III (2017) – ***Design, validation and optimizing of the adaptive Auto Landing System; comparative studies; compa-rative studies between the two designed complete ALSs***, took 9 months (January – September 2017); in this period, one has completely performed all the **9 activities**:

III.1. Optimal ALS's optimization and robustness' improvement. Study of the sensor errors and atmospheric disturbances' influence.

III.2. Adaptive components' design; neural networks' design and train (longitudinal and lateral-directional planes).

III.3. Design of the adaptive control laws (longitudinal and lateral-directional planes).

III.4. Design of the adaptive landing control systems (longitudinal and lateral-directional planes).

III.5. Interconections of the two adaptive subsystems and the obtaining of a new system for the control of aircraft landing.

III.6. Software implementation of the system for the landing adaptive control.

III.7. Adaptive ALS's optimization and robustness' improvement. Study of the sensor errors and atmospheric disturbances' influence.

III.8. Comparative studies using the two ALSs.

III.9. Dissemination of results.

One has also accomplished all the **11 specific objectives:**

OS11. To design the components of a new adaptive architecture for landing control (longitudinal plane).

OS12. To design a Pseudo Control Hedging block (PCH), for aircraft motion in longitudinal plane, by using a new method based on fuzzy logic (*innovative architecture*).

OS13. To design a new adaptive landing system (longitudinal plane) by using neural networks, the dynamic inver-sion approach, reference models, the geometry of landing, dynamic compensators, and a PCH block taking into account the sensor errors, the wind shears, and the atmospheric turbulences (*innovative architecture*).

OS14. To design the components of a new adaptive architecture for landing control (lateral plane).

OS15. To design a Pseudo Control Hedging block (PCH), for aircraft motion in lateral plane, by using a new method based on fuzzy logic (*innovative architecture*).

OS16. To design a new adaptive landing system (lateral plane) by using neural networks, the dynamic inversion approach, reference models, the geometry of landing, dynamic compensators, and a PCH block taking into account the sensor errors, the wind shears, and the atmospheric turbulences (*innovative architecture*).

OS17. To interconnect the two automatic landing subsystems and to obtain a new and innovative ALS based on the neural networks' usage and the dynamic inversion approach (*innovative auto-pilot architecture*).

OS18. To software implement, test and validate the new adaptive ALS (*software package*).

OS19. To optimize and to improve the robustness of the new adaptive ALS; to study the sensor errors and atmospheric disturbances' influence on the new optimal ALS (*optimization studies*).

OS20. To compare our new two ALSs with other already existing ALSs and to compare the new designed ALSs from the performances' point of view (*comparative studies*).

OS21. To disseminate the results in the scientific, academic and socio-economic environment.

Beside the acomplish of the scientific activities, the members of **the team achieved administrative and management activities (elaboration of scientific rapports, tasks distribution, tracking deadlines etc.), which competed at the completion of this stage in good condition.** Also, there were **regular meetings** between team members of the team, especially given that the two

PhD students involved in the project are PhD students of "Politehnica" University of Bucharest. Also, the research team was met in full in eight meetings to analyze activities and results so far achieved and to determine future actions of each member program. Funds allocated at this stage both for mobility, logistic and staff costs were managed successfully so that all objectives of the stage have been achieved. Below, it is presented the synthetic work done and results achieved in each of the nine activities intended to be conducted at this stage.

Optimal ALS's optimization/ robustness' improvement. Study of sensor errors and atmospheric disturbances' influence

Within Activity II.4. – *Design of the optimal landing control systems (longitudinal and lateral planes)* and Activity II.5. – *Interconections of the two optimal subsystems and the obtaining of a new system for the control of aircraft landing*, we designed two automatic landing subsystems (one for the longitudinal plane and other for lateral-directional plane). The obtaining of the aircraft desired landing trajectory in longitudinal plane mainly involves two variables' control: the pitch angle (θ) and the forward speed (u). According to the landing requirements for Boeing 747, the aircraft must descend from cruising altitude to a lower altitude around 420 m. Meanwhile, the aircraft speed is also reduced from the cruising speed to an approach value and, after that, it remains constant. So, when we design the desired trajectory, we design the desired forward speed u first of all. The optimal control system associated to aircraft flight during landing (longitudinal plane), based on H_2/H_∞ and dynamic inversion techniques, and the optimal observer must assure the convergences: $\Delta y = y - \bar{y} \to 0\left(y = C\mathbf{x} \to \bar{y} = C\bar{\mathbf{x}}, \mathbf{x} \to \bar{\mathbf{x}}\right), \Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}} \to 0 \ \left(\hat{\mathbf{x}} \to \mathbf{x} \to \bar{\mathbf{x}}\right)$; here, $\bar{\mathbf{x}}$ is aircraft desired state, while $\bar{y}$ is the reference vector associated to the measured output y.

In figs. 3.1 and 3.2 we represent the time characteristics for the glide slope phase and flare phase, respectively; the characteristics have been represented for the new ALS affected by disturbances in the presence or in the absence of sensor errors (the sensors are used for the measurement of the states). The last three mini-graphics in these two figures represent the deviations of the forward speed (u), slope angle (γ), and altitude (H), with respect to their nominal values, i.e. $\bar{u} - u, \gamma_c - \gamma, H_{ref} - H$. The presence of the sensor errors is not visible: the curves with solid line (obtained for the ALS without sensor errors) overlap almost perfectly over the curves plotted with dashed line (obtained for the ALS with sensor errors). The time origin for the flare trajectory is chosen zero when the altitude is H=H_0=30 m (the altitude at which the glide slope phase ends) [1].

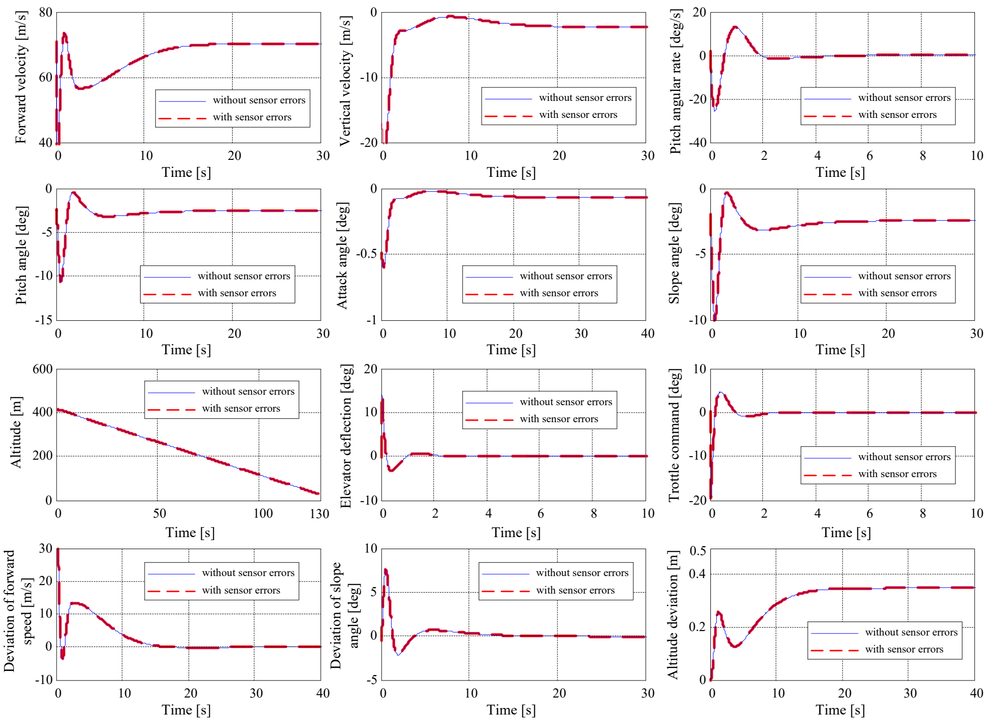


Fig. 3.1. Time characteristics for the glide slope phase, with or without sensor errors

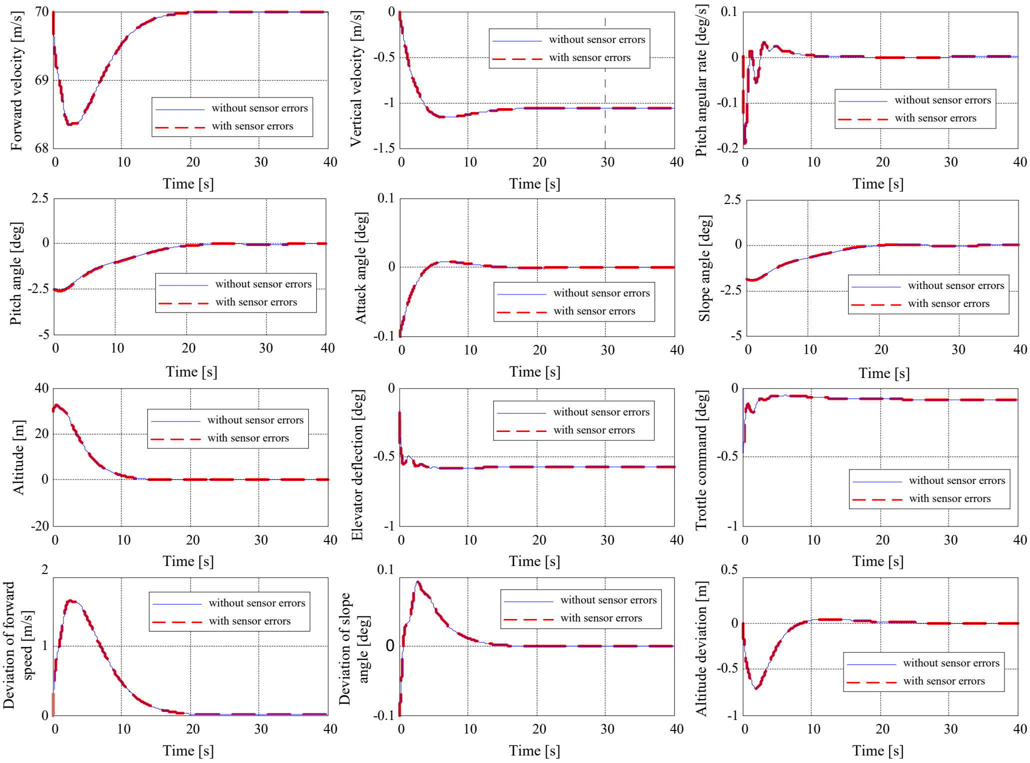


Fig. 3.2. Time characteristics for the flare phase, with or without sensor errors

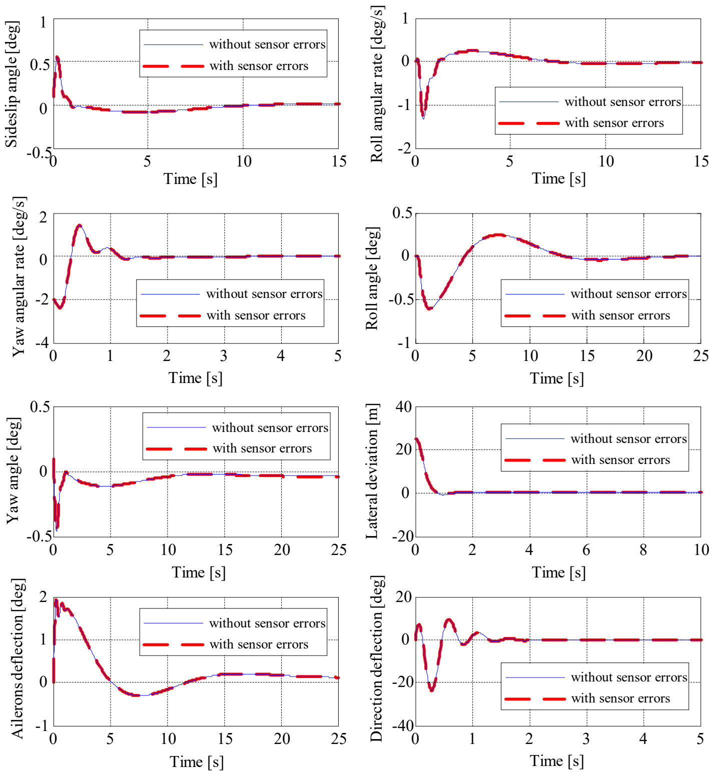


Fig. 3.3. Time characteristics of the lateral-directional control system, with linear actuators, with or without sensor errors

In Fig. 3.3 we represented the time characteristics for the flight direction control subsystem; the characteristics have been represented for the second control system affected by low crosswind (V_{vy}=2 m/s), with linear actuators (v_h=0), in the presence or in the absence of sensors' errors. The same conclusion can be drawn regarding the overlap of the curves with solid line and the curves plotted with dashed line. We also tested the lateral subsystem, with linear actuators (v_h=0), for different values of the crosswind (between 2 m/s and 10 m/s) in order to analyze the system's robustness when the lateral wind has medium or high values. Thus, in Fig. 3.4 we represented the time history of the main two control variables (Y and β) for different values of the lateral wind. Analyzing Fig. 3.4, one can conclude that the errors Y-Y_c and β-β$_c$ increase together with the value of the crosswind; for example, the error with respect to Y is 0.58 m for V_{vy}=2 m/s, 1.21 m for V_{vy} =4 m/s, 1.85 m for V_{vy} =6 m/s, 2.50 m for V_{vy} =8 m/s, and 3.14 m for V_{vy} =10 m/s, respectively; all the errors meet the FAA accuracy requirements for Category III, although the lateral wind take even large values. On the other hand, the errors with respect to β, for different values of the lateral wind, are very small (less than -0.02 deg). It is very important to remark that all these errors (both for Y and β) are smaller than their homologues in the case of the first ALS [2].

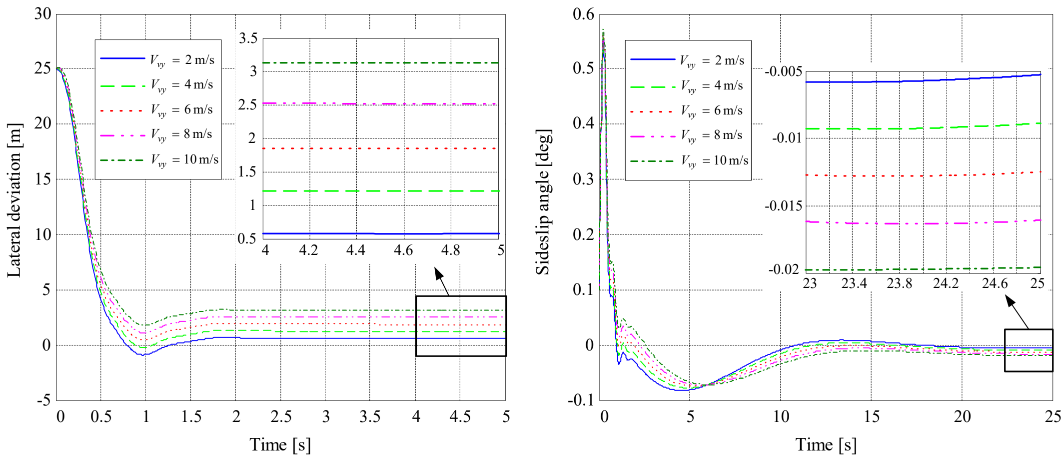


Fig. 3.4. Time history of the main two control variables for different values of the lateral wind (ALS with linear actuators)

Geometry of the landing process in longitudinal and lateral-directional planes

If only the longitudinal plane approach is considered for the landing of an aircraft, then two phases are distinguished for this procedure (Fig. 3.5): 1) ***Glide slope*** phase ($H \geq H_0$), and 2) ***Flare*** phase ($H < H_0$); H is the aircraft altitude and H_0 is the starting altitude for the flare landing phase; $A_p(X_p, H_p)$ is the starting point for the glide slope phase (the segment $\overline{A_p A_0}$), $A_0(X_0, H_0)$– the starting point for the flare phase (the segment $\overline{A_0 A_{td}}$), $A_{td}(X_{td}, 0)$– the point where the aircraft touches the runway, $\gamma_c = -2.5\,\text{deg}$– the calculated (desired) value of aircraft glide slope angle, X– the horizontal coordinate associate to the displacement, $H_c = H_c(X)$– the calculated flight altitude, and V_0– aircraft nominal velocity.

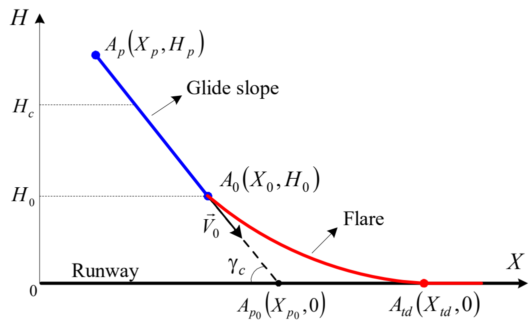


Fig. 3.5. Aircraft landing geometry in longitudinal plane

The equations describing the aircraft landing geometry in longitudinal plane are [3]:

$$X_{p_0} = X_p - H_p / \tan(\gamma_c), \tag{3.1}$$

$$H_c = \begin{cases} (X - X_{p_0}) \cdot \tan(\gamma_c), \text{for } H \geq H_0, \\ H_c = H_0 \exp((X_0 - X)/\tau\dot{X}) \Leftrightarrow \dot{H}_c = -H_c/\tau, \text{for } H < H_0, \end{cases} \tag{3.2}$$

with $\dot{X}$– aircraft horizontal forward velocity and τ– the time constant that defines the exponential curvature (flare landing phase). If the flare process takes, for example, 5τ seconds [4] and the velocity of the aircraft has not a significant variation, then, $X_{td} - X_0 = V_0 \cdot 5\tau$. To calculate the value of τ, the following expression is used [3]:

$$V_0 \tau = 0.9741(X - X_{p_0}). \tag{3.3}$$

Indeed, removing $\dot{H}_0$ between the second equation (3.2) for $H = H_c = H_0$ and the equation which results from the velocities triangle (fig. 3.5), one gets: $\dot{H}_0 = -\frac{H_0}{\tau} = V_0 \sin\gamma_c \cong -\frac{2.5}{180/\pi} V_0$. Thus, $H_0 = \frac{2.5}{57.3} V_0 \tau$, with H_0 having the form which results from the equation: $H_0 = -(X_{p_0} - X_0)\tan\gamma_c = 0.0435 \cdot (X_{p_0} - X_0)$ and, removing H_0 between the last two equations, one obtains the form (3.3).

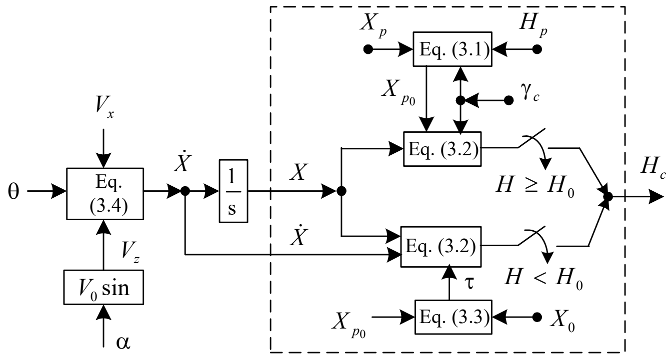


Fig. 3.6. The block diagram of aircraft landing geometry in longitudinal plane

Thus, the geometry of aircraft motion in longitudinal plane, during landing, is described by the equation:

$$\dot{X} = V_x \cos\theta + V_z \sin\theta, V_z = V_0 \sin\alpha, \tag{3.4}$$

where V_x and V_z are the aircraft velocity components along aircraft longitudinal axis and normal axis, $\theta -$ the pitch angle, while α is the attack angle.

The equations (3.1)-(3.4) describe the block diagram of aircraft landing geometry in longitudinal plane.

Design of the new adaptive ALS (longitudinal plane)

The new landing subsystem for longitudinal plane contains: 1) a subsystem for calculation of the variable H_c; 2) reference models; 3) dynamic inversion and neural network based adaptive controller; 4) a PCH block which makes the system work in the nonlinearities' linear zones. The reference models are used for obtaining the calculated altitude and the desired flight velocity. The first automatic landing subsystem contains an adaptive system for the control of the output vector $y = [H \quad V_x]^T$. The structure of the adaptive control subsystem (fig. 3.7) has on its direct way of the closed loop having unitary feedback an adaptive controller and a system (A+ACT) consisting of the nonlinear model (having the function h_r) and the reduced order linear model (having the transfer matrix $H_d(s)$). The control law $\hat{v}$ may be chosen of the following form [5, 6]:

$$\hat{v} = v_{pd} + \hat{v}_r - v_a + \bar{v}, \tag{3.5}$$

where the signal v_{pd} (output of a linear dynamic compensator) is used for the stabilization of airplane linear dynamics, $v_a -$ an adaptive component (output of a neural network – NN$_c$) is useful to compensate the nonlinear function $h_r(v, y)$ approximation error (ε), functions interfering in the airplane and actuators' dynamics, $\hat{v}_r$ (signal provided by the reference models) is used to cancel the system's deviation $\tilde{y}$ and its derivatives, while $\bar{v}$ is a robustness component. The state of the longitudinal dynamics is $\boldsymbol{x} = [V_x \quad \alpha \quad \omega_y \quad \theta \quad H]^T$ [5]; $\omega_y = \dot{\theta}$ is the pitch rate. The command vector of the first subsystem is chosen as: $u = [\delta_p \quad \delta_T]^T$, where δ_p is the elevator deflection and $\delta_T -$ the thrust command; their equations are:

$$T_p \dot{\delta}_p + \delta_p = \delta_{p_c}, T_T \dot{\delta}_T + \delta_T = \delta_{T_c}, \tag{3.6}$$

with δ_{p_c} and δ_{T_c} – the commands applied to elevator and to engines, respectively; T_p and T_T are time constants. If the servo-elevator is described by a nonlinear dynamics, the first equation (3.6) is replaced by the model from fig. 3.8 (with $\hat{v}_i = \hat{v}_1, \delta_i = \delta_p$, $T_i = T_p, v_{h_i} = v_{h_1}$). Because, the adaptive controllers are sensitive to actuators' nonlinearities, in these controllers' architectures, one should introduce Pseudo Control Hedging (PCH) blocks which limit the signal v by means of a component representing the estimation of the actuators' dynamics. The output signal of the PCH block (v_{h_1}) is a reference model's additional input.

The ensemble airplane-actuators (A+ACT) is described by equation [7]: $\dot{\boldsymbol{x}} = A_v \boldsymbol{x} + B_v u$, with $\boldsymbol{x} = [V_x \ \alpha \ \omega_y \ \theta \ H \ \delta_p \ \delta_T]^T$, $u = [\delta_{p_c} \ \ \delta_{T_c}]^T$; the expressions of A_v and B_v are presented in [5]. The system A+ACT is generally described by state equations: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, u), y = h(\boldsymbol{x})$, with f and h – generally unknown nonlinear/linear functions, calculated in the linearization process with realtion to the landing trajectory.

The relative degrees of the system with respect to the variables y_i are denoted with $r_i, i = \overline{1,2}; r = r_1 + r_2$ is the relative degree of the system with respect to the output vector. The control law $\hat{v} = \hat{h}_r(y, \hat{u})$ represents the best approximation of the function $v = h_r(y, u)$; the approximation error of the function $h_r(y, u)$ is $\varepsilon = [\varepsilon_1 \ \ \varepsilon_2]^T$ having the following form: $\varepsilon = h_r(y, u) - \hat{h}_r(y, \hat{u})$. In [5] one has obtained $r_1 = 3, r_2 = 2, r = 5, \begin{bmatrix} \dddot{H} + \ddot{H} \\ \ddot{V}_x \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & 0 & c_p & c_t \\ d_1 & d_2 & d_3 & d_4 & 0 & d_p & d_t \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \dfrac{a_{52} b_{21}}{T_p} & 0 \\ 0 & \dfrac{b_{12}}{T_T} \end{bmatrix} \begin{bmatrix} \delta_{p_c} \\ \delta_{T_c} \end{bmatrix}$; it can be written: $y^{(r)} = -\lambda^T Y + v, v = \hat{v} + \varepsilon$, with $\lambda = [\lambda_1 \ \ \lambda_2]^T, Y = [Y_1 \ \ Y_2]^T, h_r = [h_{r_1} \ \ h_{r_2}]^T$. By identification, one gets [5]:

$$\hat{u} = \begin{bmatrix} \hat{\delta}_{p_c} \\ \hat{\delta}_{T_c} \end{bmatrix} = \hat{h}_r^{-1}(y, \hat{v}) = \begin{bmatrix} \dfrac{a_{52} b_{21}}{T_p} & 0 \\ 0 & \dfrac{b_{12}}{T_T} \end{bmatrix}^{-1} \left\{ \hat{v} - \begin{bmatrix} c_1 \\ d_1 \end{bmatrix} V_x \right\}, \tag{3.7}$$

with $\hat{v} = \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} = \begin{bmatrix} c_2 & c_3 & c_4 & c_p & c_t \\ d_2 & d_3 & d_4 & d_p & d_t \end{bmatrix} [\alpha \ \ \omega_y \ \ \theta \ \ \delta_p \ \ \delta_T]^T$. Using the expressions for reduced order linear subsystems' transfer functions, one uses the reference models for y_1 and y_2; the block diagrams of the blocks modelling these reference models are presented in Fig. 3.9. Beside $\overline{y}_i$, the reference models provide the components $\hat{v}_{r_i}, i = \overline{1,2}$, of the vector $\hat{v}_r = [\hat{v}_{r_1} \ \ \hat{v}_{r_2}]^T$. This vector is obtained from the condition that the vector y and its derivatives tend to their imposed values, i.e. $y^{(r)} = \overline{y}^{(r)}$ and $Y = \overline{Y}$, which is obtained in steady regime when $v_a = \varepsilon$ and $v = \hat{v}$; this means that the approximation error is compensated by the adaptive component; from equation $y^{(r)} = -\lambda^T Y + v, v = \hat{v}$, one obtains: $\hat{v}_r = \overline{y}^{(r)} + \lambda^T \overline{Y}$, with $\lambda = [\lambda_1 \ \ \lambda_2]^T = [0 \ \ 0 \ \ 1 \ \ 0 \ \ 0]^T$. Thus, $\hat{v}_{r_1} = \dddot{\overline{y}}_1 + \ddot{\overline{y}}_1$ and $\hat{v}_{r_2} = \ddot{\overline{y}}_2$. Moreover, from (3.7), omitting the term which contains δ_{T_c}, one gets: $\hat{v}_1 = c_1 V_x + \dfrac{a_{52} b_{21}}{T_p} \hat{\delta}_{p_c}$, $\hat{\delta}_{p_c} = \dfrac{T_p}{a_{52} b_{21}} (\hat{v}_1 - c_1 V_x)$. By substituting $\hat{v}_1$ with $\overline{v}_1$ and $\hat{\delta}_{p_c}$ with $\overline{\delta}_p$ in the previous equation and $\hat{\delta}_{p_c}$ with δ_{p_c}, the functions $\hat{h}_{r_1}$ and $\hat{h}_{r_1}^{-1}$ from fig. 3.8 are deduced [1]:

$$\overline{v}_1 = c_1 V_x + \frac{a_{52} b_{21}}{T_p} \overline{\delta}_p, \delta_{p_c} = \hat{h}_{r_1}^{-1}(y, \hat{v}_1) = \frac{T_p}{a_{52} b_{21}} (\overline{v}_1 - c_1 V_x). \tag{3.8}$$

The two components of vector v_{pd} are: $v_{pd_1} = k_{p_1} \tilde{y}_1 + k_{d_1} \dot{\tilde{y}}_1 = [k_{p_1} \ \ k_{d_1} \ \ 0] \cdot [\tilde{y}_1 \ \ \dot{\tilde{y}}_1 \ \ \ddot{\tilde{y}}_1]^T = d_{c_1} \boldsymbol{e}_1, v_{pd_2} = k_{p_2} \tilde{y}_2 + k_{d_2} \dot{\tilde{y}}_2 = [k_{p_2} \ \ k_{d_2}]$ $[\tilde{y}_2 \ \ \dot{\tilde{y}}_2]^T = d_{c_2} \boldsymbol{e}_2$, with $\boldsymbol{e}_1 = [\tilde{y}_1 \ \ \dot{\tilde{y}}_1 \ \ \ddot{\tilde{y}}_1]^T$ – the vector of errors with respect to the state y_1 associated to the reduced order subsystem having $r_1 = 3$, while $\boldsymbol{e}_2 = [\tilde{y}_2 \ \ \dot{\tilde{y}}_2]^T$ – the error state vector of the system with respect to the state y_2 associated to the other reduced order subsystem $(r_2 = 2)$. The notations $d_{c_1} = \lfloor k_{p_1} \quad k_{d_1} \quad 0 \rfloor$ and $d_{c_2} = \lfloor k_{p_2} \quad k_{d_2} \rfloor$ are used; k_{p_i} and k_{d_i} are the compensator's coefficients; one obtains the vector [5]: $v_{pd} = \begin{bmatrix} v_{pd_1} \\ v_{pd_2} \end{bmatrix} = \begin{bmatrix} d_{c_1}\boldsymbol{e}_1 & d_{c_2}\boldsymbol{e}_2 \end{bmatrix}^T = \begin{bmatrix} d_{c_1} & 0_{1\times 2} \\ 0_{1\times 3} & d_{c_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = D_c\boldsymbol{e}$.

The linear observer estimates the state vector $\boldsymbol{e}$ providing $\hat{\boldsymbol{e}}$ (estimation of the vector $\boldsymbol{e}$); the observer uses only the input vector: $z_c = \tilde{y} = [\tilde{y}_1 \quad \tilde{y}_2]^T = [c_1\boldsymbol{e}_1 \quad c_2\boldsymbol{e}_2]^T = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix}\boldsymbol{e} = \overline{C}\boldsymbol{e}$, with $C = \begin{bmatrix} c_1 & 0_{1\times 3} \\ 0_{1\times 2} & c_2 \end{bmatrix}$, $c_1 = [1 \quad 0 \quad 0], c_2 = [1 \quad 0]$. The observer is modeled by the equations [5, 8]:

$$\dot{\hat{\boldsymbol{e}}} = \tilde{A}\hat{\boldsymbol{e}} + L z_c \,, \tilde{A} = \overline{A} - L\overline{C} \,, \tag{3.9}$$

where L is the observer's gain matrix which is calculated in order to obtain an asymptotically stable matrix $\tilde{A}$. For the obtaining of the adaptive control law, one considers a feed-forward neural network (NN_c) consisting of 3 layers: an input layer (with n_1 neurons), a hidden layer (with n_2 neurons), and an output layer (with n_3 neurons); W denotes the matrix consisting of the weights associated to the links between the hidden layer's neurons and output layer's neurons and V – the matrix consisting of the weights associated to the links between the input layer neurons and the hidden layer neurons. The adaptive control law is [9]:

$$v_a = W^T \sigma\left(V^T \eta\right), \tag{3.10}$$

where the matrices W and V, associated to the NN_c's weights, are the solutions of the equations' system [10]:

$$\begin{cases} \dot{W} = -k_W\left[2\left(\sigma - \sigma' V^T \eta\right)\overline{\boldsymbol{e}} + k\left(W - W_0\right)\right], \\ \dot{V} = -k_V\left[2\eta\overline{\boldsymbol{e}}W^T\sigma' + k\left(V - V_0\right)\right], \end{cases} \tag{3.11}$$

with $\overline{\boldsymbol{e}} = \hat{\boldsymbol{e}}^T P\overline{B}\,; \sigma' = \left.\dfrac{\mathrm{d}\sigma}{\mathrm{dz}}\right|_{z=z_0}$ – Jacobian of the vectorial sigmoid function σ, k_W, k_V – positive constants; the expressions for obtaining k is presented in detail in [5]. The vector η is:

$$\eta = \begin{bmatrix} 1 & \hat{v}_d^T & y_d^T \end{bmatrix}^T = \begin{bmatrix} 1 & I_1 & I_2 & \ldots & I_{n1} \end{bmatrix}^T; \tag{3.12}$$

$\hat{v}_d^T = [\hat{v}(t) \quad \hat{v}(t-d) \quad \ldots \quad \hat{v}(t-(n_1-r-2)d)]$, d – the sample step, $y_d^T = [y(t) \quad y(t-d) \quad \ldots \quad y(t-(n_1-r-3)d)]$, while $I_i, i=\overline{1,n_1}$ are the neural network's outputs. The expression $\overline{v}$ (the robustness component of the control law (3.5)) is borrowed from [5] or [9]; the next equation is used:

$$\overline{v}^T = k_z\left(\|F\|_f + \overline{F}\right)\|\hat{\boldsymbol{e}}\|\frac{\overline{\boldsymbol{e}}}{\|\overline{\boldsymbol{e}}\|} + k_e\overline{\boldsymbol{e}}\,, \tag{3.13}$$

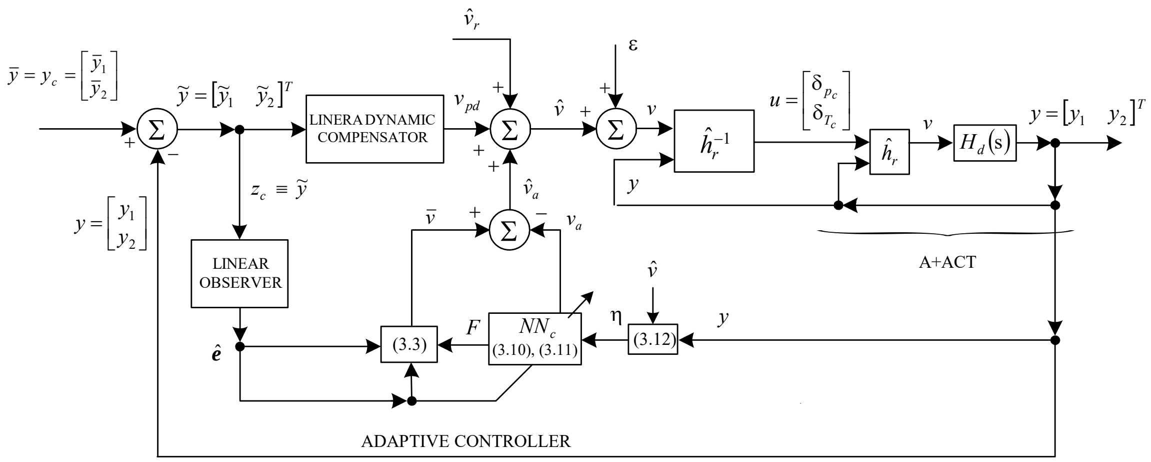


Fig. 3.7. The adaptive control subsystem of the output vector y

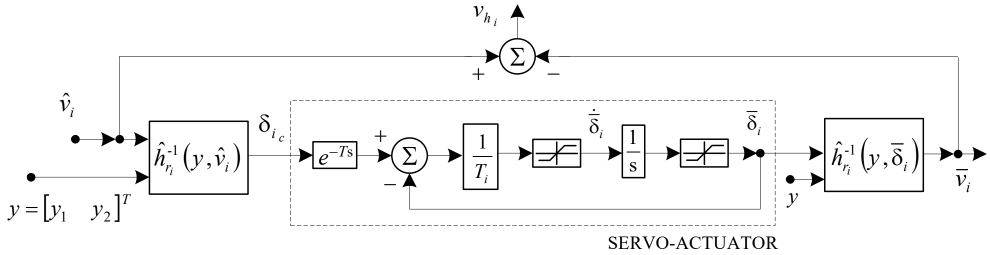


Fig. 3.8. Block diagram associated to the actuators and to the PCH blocks

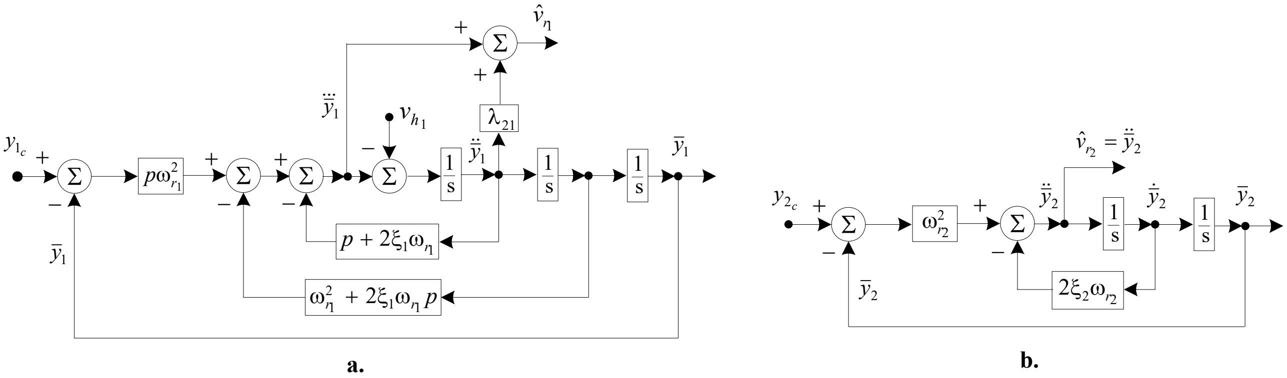


Fig. 3.9. Block diagrams of the reference models: a) r_1=3 and b) r_2=2

with k_z and k_e – positive gains, $F=\begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}; \|F\|_f^2 = \mathrm{tr}\left\{F^T F\right\} \le \overline{F}; \|F\|_f$ is the Frobenius norm of the matrix F, while $\overline{F}$ is the neural network's ideal matrix. The role of the robustness component $\bar{\bar{v}}$ is to ensure the boundedness of the neural network's matrices W and V.

The controllers may be sensitive to actuators' nonlinearities; therefore, in their architectures, one may introduce a block which limits the pseudo-control by means of a component representing an estimation of the execution element's dynamics (PCH - Pseudo Control Hedging). The input saturation and the input rate saturation may also be significant problems. For lateral-directional plane, we introduce a PCH block which limits the signal v with a component representing the actuator dynamics' estimation. Thus, PCH "moves back" the reference model, introducing a reference model's response correction with respect to the estimation of the execution element's position. The PCH blocks are more useful in the case of neural networks' usage; the strong point of the neural networks is their approximation ability, these being capable to approximate an unknown system dynamics through learning; also, the PCH blocks eliminate the NNs' adapting difficulties. The main purpose of the Pseudo Control Hedging block is to prevent the adaptive element of a control system from trying to adapt to a class of system input characteristics (characteristics of the plant or of the controller).

Design of the adaptive ALS (lateral-directional plane)

For the lateral-directional plane, the landing geometry is detailed in [11]; the second main subsystem of the new designed ALS must cancel airplane's lateral deviation relative to the runway; its main control subsystem is again the adaptive system presented in Fig. 3.7. The lateral dynamics associated to the state is $\boldsymbol{x}=\begin{bmatrix} V_y & \omega_x & \omega_z & \varphi & \Delta\psi \end{bmatrix}^T$ with V_y – the airplane's lateral velocity, ω_x and ω_z – the roll and the yaw angular rates, φ – the roll angle, $\Delta\psi$ – the variation of the flight direction angle; δ_e and δ_d are the actuators' states (δ_e – ailerons and δ_d – rudder's deflections). The linear equations of the actuators (servo-aileron and servo-rudder) are [3]:

$$\begin{aligned} T_e\dot{\delta}_e + \delta_e &= \delta_{e_c}, \\ T_d\dot{\delta}_d + \delta_d &= \delta_{d_c}, \end{aligned} \tag{3.14}$$

or the ones describing the nonlinear subsystems of the PCH models in Fig. 3.8; for $i=e, \delta_{i_c}=\delta_{e_c}$ is the command of the aileron's actuator, while, for $i = d$, $\delta_{i_c} = \delta_{d_c}$ is the command of the rudder associated actuator. For the system containing the lateral-directional dynamics and the actuators' linear or nonlinear dynamics is $u = \begin{bmatrix} \delta_{e_c} & \delta_{d_c} \end{bmatrix}^T$. By using the same procedure as in section II, the relative degrees r_1 and r_2 are obtained: $r_1 = r_2 = 3, r = 6$. One also obtains [3]:

$$\hat{u} = \begin{bmatrix} \hat{\delta}_{e_c} \\ \hat{\delta}_{d_c} \end{bmatrix} = \hat{h}_r^{-1}(y, \hat{v}) = \begin{bmatrix} \dfrac{b_{21}}{T_e} & \dfrac{b_{22}}{T_d} \\ \dfrac{b_{31}}{T_e} & \dfrac{b_{32}}{T_d} \end{bmatrix}^{-1} \left\{ \hat{v} - \begin{bmatrix} c_4 \\ d_4 \end{bmatrix} \varphi \right\}, \tag{3.15}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & c_e & c_d \\ d_1 & d_2 & d_3 & d_e & d_d \end{bmatrix} \begin{bmatrix} V_y & \omega_x & \omega_z & \delta_e & \delta_d \end{bmatrix}^T. \tag{3.16}$$

The control law *u* has the form (3.15), where $\hat{v}$ is replaced with v and $\hat{\delta}$ with δ. The reference models are the ones in Fig. 3.9.a. The signals v_{h_i} from this figure are the components of the vector v_h – the output of the PCH nonlinear consisting of two subsystems having the architectures in Fig. 3.8. The PCH block is introduced when the servo-aileron and servo-rudder are nonlinear, issue affecting the neural networks' functioning. Using again the above presented methodology, one also obtains:

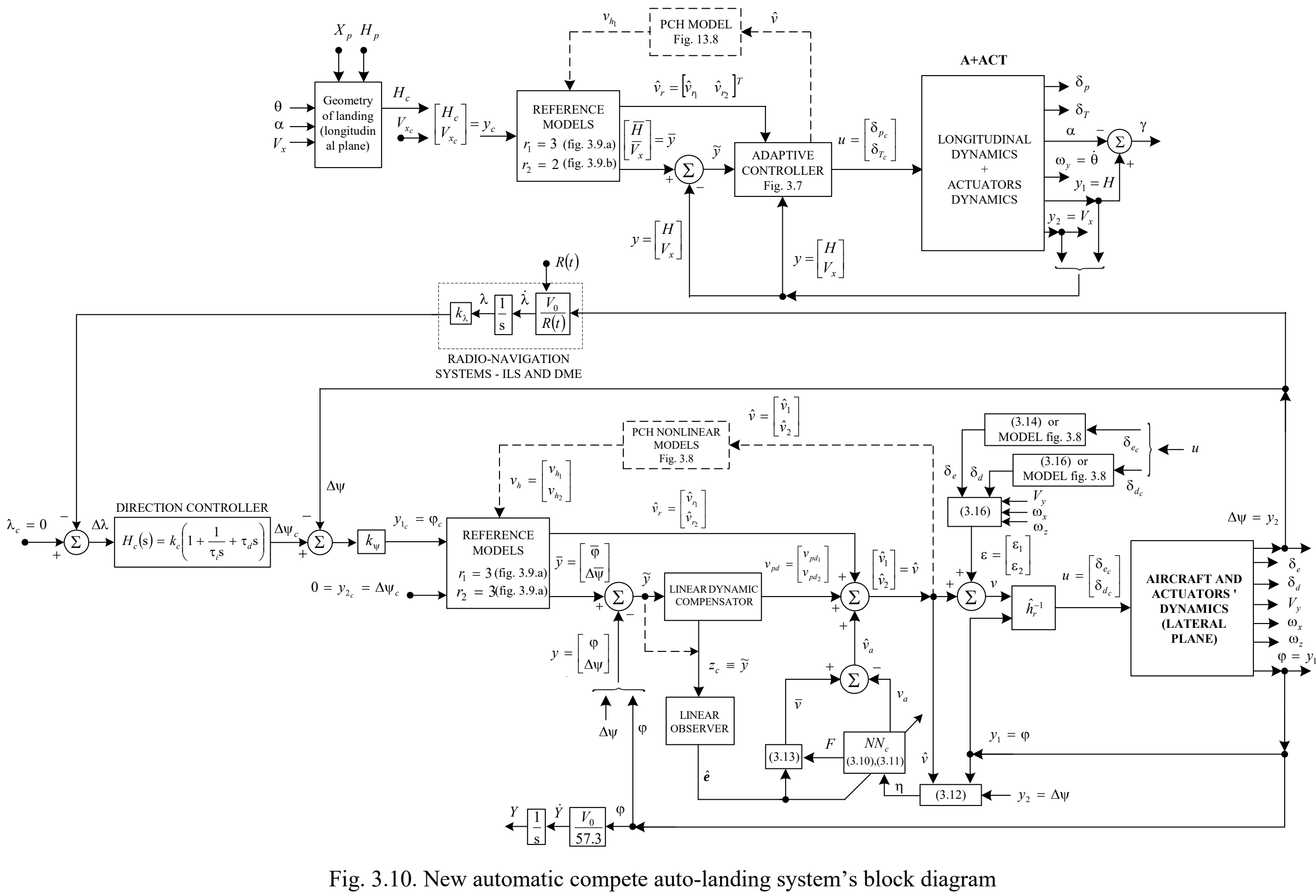


Fig. 3.10. New automatic compete auto-landing system's block diagram

$$\hat{v}_1 = c_4\varphi + \frac{b_{21}}{T_e}\hat{\delta}_{e_c}, \hat{v}_2 = d_4\varphi + \frac{b_{32}}{T_d}\hat{\delta}_{d_c}, \bar{v}_1 = \hat{h}_{r_1}(y, \bar{\delta}_1) = \hat{h}_{r_1}(y, \bar{\delta}_e) = c_4\varphi + \frac{b_{21}}{T_e}\bar{\delta}_e, \bar{v}_2 = \hat{h}_{r_2}(y, \bar{\delta}_2) = \hat{h}_{r_2}(y, \bar{\delta}_d) = d_4\varphi + \frac{b_{32}}{T_d}\bar{\delta}_d.$$

For the lateral-directional plane, the system has the state vector $\boldsymbol{e} = \begin{bmatrix} \tilde{y} & \dot{\tilde{y}} & \ddot{\tilde{y}} \end{bmatrix}^T = \begin{bmatrix} e & \dot{e} & \ddot{e} \end{bmatrix}^T = \begin{bmatrix} \tilde{y}_1 & \dot{\tilde{y}}_1 & \ddot{\tilde{y}}_1 & \tilde{y}_2 & \dot{\tilde{y}}_2 & \ddot{\tilde{y}}_2 \end{bmatrix}^T$. If a PD linear dynamic compensator is chosen, its state is $\boldsymbol{e} = \begin{bmatrix} e_1 & e_2 \end{bmatrix}^T = \tilde{y} = \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 \end{bmatrix}^T = \begin{bmatrix} \tilde{\varphi} & \Delta\tilde{\psi} \end{bmatrix}^T$, with $\tilde{\varphi} = \bar{\varphi} - \varphi$ and $\Delta\tilde{\psi} = \Delta\bar{\psi} - \Delta\psi$. The expression of the control law v_{pd} is [11]:

$$v_{pd} = \begin{bmatrix} d_{c_1}\boldsymbol{e}_1 & d_{c_2}\boldsymbol{e}_2 \end{bmatrix}^T = D_c\begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix}^T = \begin{bmatrix} k_{p_1} & k_{d_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{p_2} & k_{d_2} & 0 \end{bmatrix}\boldsymbol{e}. \tag{3.17}$$

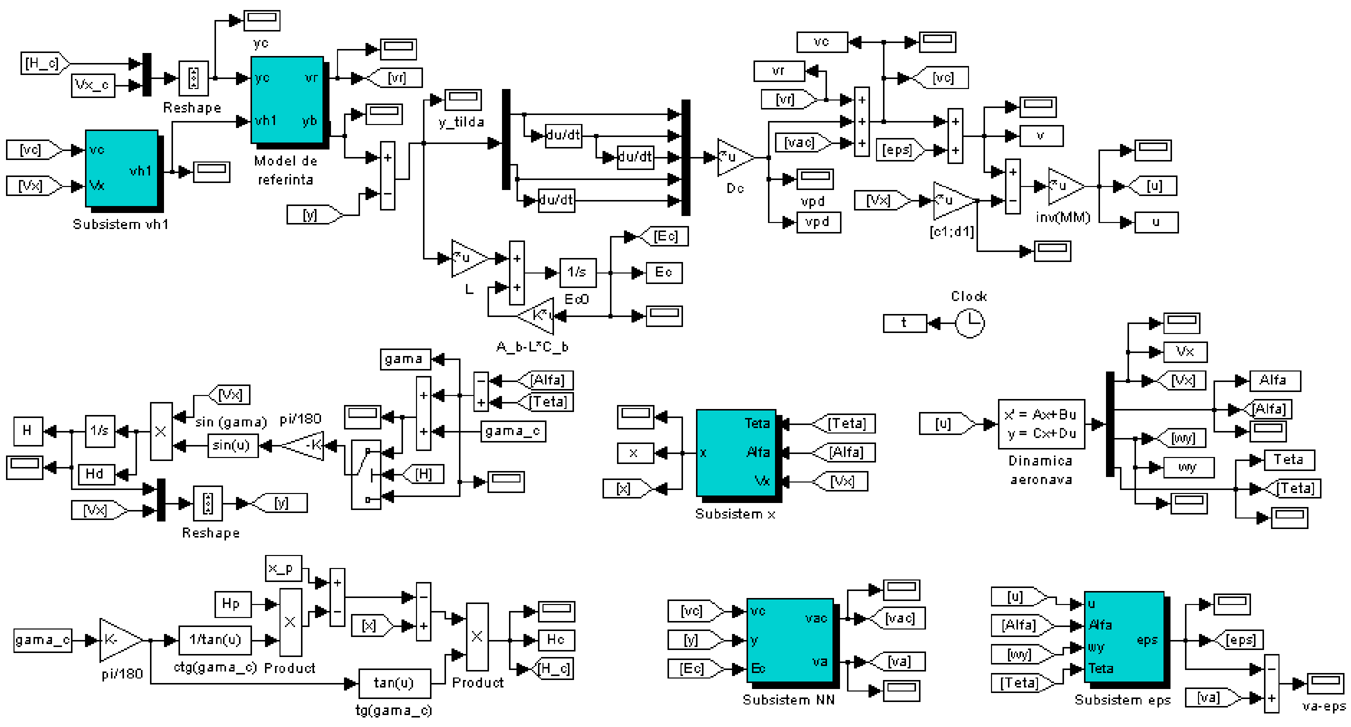
Because in steady regime $\hat{v} = \varepsilon = v = 0$, one obtains the error model's equation: $\dot{\boldsymbol{e}} = A\boldsymbol{e} - Bv$, $v = v_{pd} + \hat{v}_r + \hat{v}_a + \varepsilon$, with $\boldsymbol{e} = \bar{Z} - Z = [\boldsymbol{e}_1 \quad \boldsymbol{e}_2]^T$. Replacing in previous eqs. $v_{pd} = D_c\boldsymbol{e}$, one gets: $\dot{\boldsymbol{e}} = \bar{A}\boldsymbol{e} - \bar{B}(\hat{v}_r + \hat{v}_a + \varepsilon), \hat{v}_a = \bar{v} - v_a, \bar{B} = B, \bar{A} = A - BD_c$ [11].

The structure of the new neural network based ALS, using dynamic inversion, is presented in Fig. 3.10; it consists of two subsystems – one for longitudinal plane and one for the lateral-directional plane. For the first one, the airplane's speed (V_x) and altitude (H) are controlled in order to track the desired trajectory. In lateral plane, there are controlled the airplane's roll angle (φ) and the error angle $\Delta\psi = \bar{\psi} - \psi$ (difference between $\bar{\psi}-$ runway's direction and $\psi-$ airplane's flight direction).

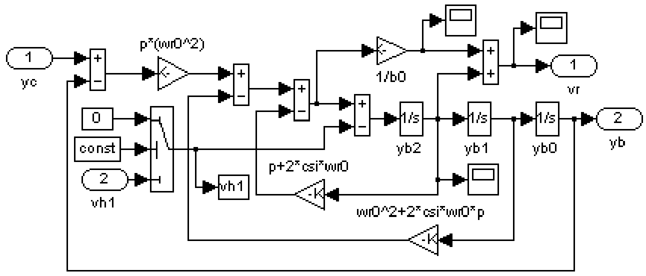
Software implementation of the system for the landing adaptive control

To study the functionality of the new ALS, one considers a light airplane (Charlie type) [7]; the coefficients associated to its dynamics (both planes) are given in [5] and [11]. The reference models' coefficients have the following values: $p = 0.5, \xi_1 = 0.7$, $\omega_{r_1} = 2.5\,\text{rad/s}, \omega_{r_2} = 2.5\,\text{rad/s}, \xi_2 = 0.7$, while the control limits of the servo-actuators are $\pm 5\,\text{deg}$ and $\pm 5\,\text{deg/s}$, respectively. For longitudinal plane, using the linear subsystems' characteristic equations (having unitary negative feedbacks) having the outputs $\bar{y}_1$ and $\bar{y}_2$, the values of the dynamic compensators' parameters are calculated; imposing for the characteristics equations $s^3 + s^2 + k_{d_1}s + k_{p_1} = 0, s^2 + k_{d_2}s + k_{p2} = 0$, the roots -0.1, -0.2, -0.3 and -1.1, -0.7, respectively, one gets: $k_{p_1} = 0.02, k_{d_1} = 0.27$, $k_{p_2} = 0.77, k_{d_2} = 0.8$. For lateral-directional plane, imposing for characteristics eqs. $s^3 + s^2 + k_{d_i}s + k_{p_i} = 0, i = \overline{1,2}$, negative roots (e.g. -0.1, -0.2, -0.3 and -1.1, -0.7, -0.6), it results: $k_{p_1} = 0.0193, k_{d_1} = 0.27, k_{p_2} = 0.77, k_{d_2} = 0.8$.

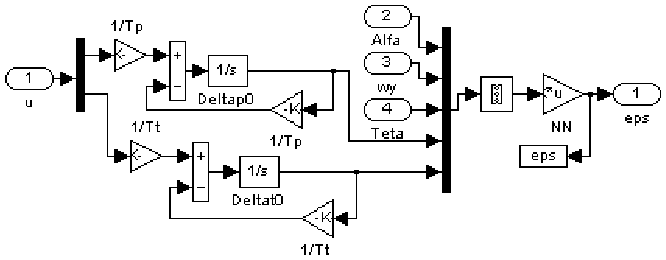
The Matlab/Simulink model for the first phase of landing in longitudinal plane is presented in Fig. 3.11.a with the subsystems: b) „reference model (model de referinţă)"; c) „subsystem eps (subsistem eps)"; d) „subsystem vh1 (subsistem vh1)" (for the calculation of the nonlinear model PCH's output; e) „subsystem X (subsistem X)" (for the calculation of the horizontal coordinate X); f) „subsystem NN (subsistem NN)" (for the calculation of the components v_a and $\bar{v}$ of the adaptive command $\hat{v}_a$), with its two subsystems: g) „subsystem ita (subsistem ita)" (for the calculation of the vector η) and h) „subsystem NNc (subsistem NNc)" (for the calculation of the output components of the NNc). „Subsystem NNc (Subsistemul NNc)" with its two subsystems: i) „subsystem differnetial equations (subsistem ecuaţii diferenţiale)"; j) „subsystem sigmap (subsistem sigmap)" (for calculation of the matrix σ').



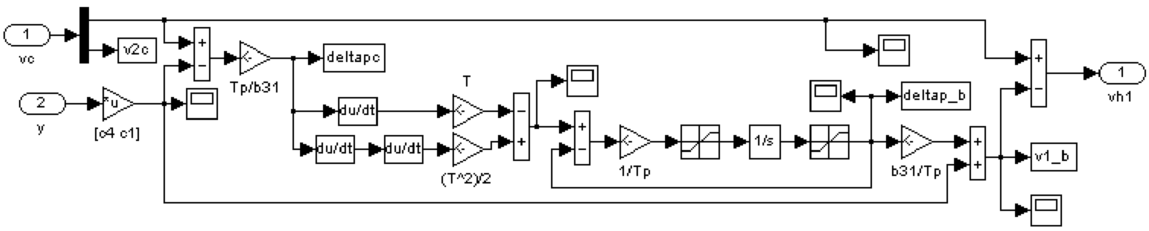
a.


yc
p*(wr0^2)
1/b0
vr
yb
const
vh1
p+2*csi*wr0
1/s
yb2
yb1
yb0
wr0^2+2*csi*wr0*p

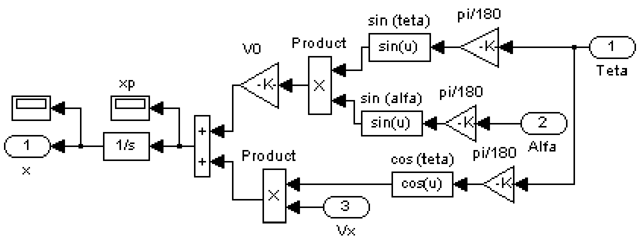
b.


u
1/Tp
1/s
Deltap0
1/Tt
Deltat0
Alfa
wy
Teta
NN
eps

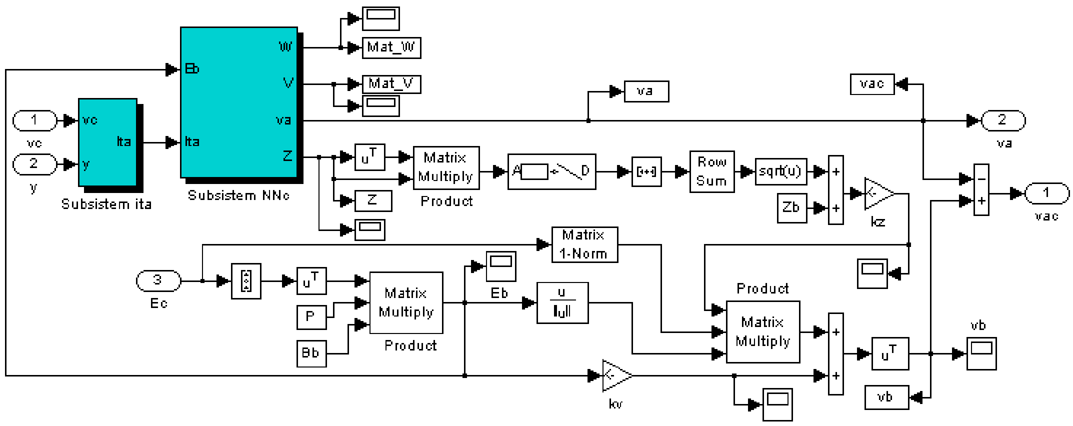
c.


vc
v2c
y
[c4 c1]
Tp/b31
deltapc
du/dt
T
(T^2)/2
1/Tp
1/s
deltap_b
b31/Tp
vh1
v1_b

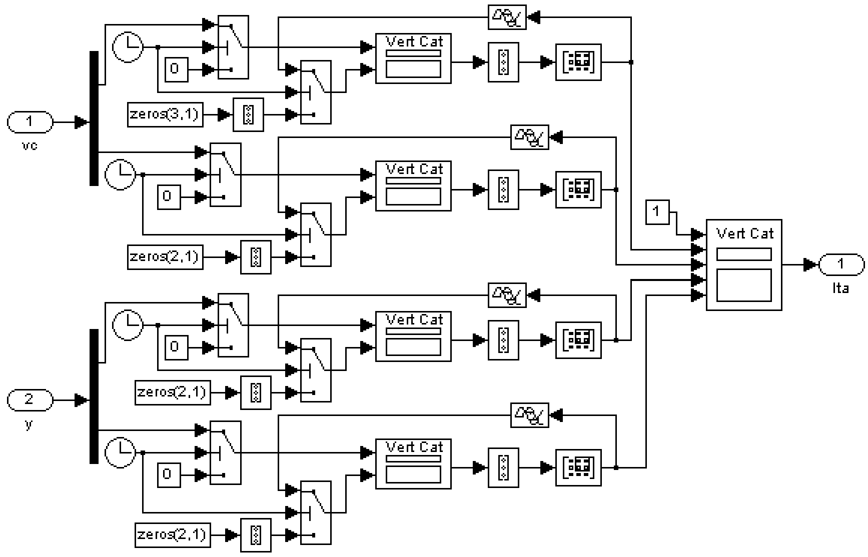
d.


x
xp
1/s
V0
Product
sin (teta)
sin(u)
pi/180
Teta
sin (alfa)
Alfa
cos (teta)
cos(u)
Vx

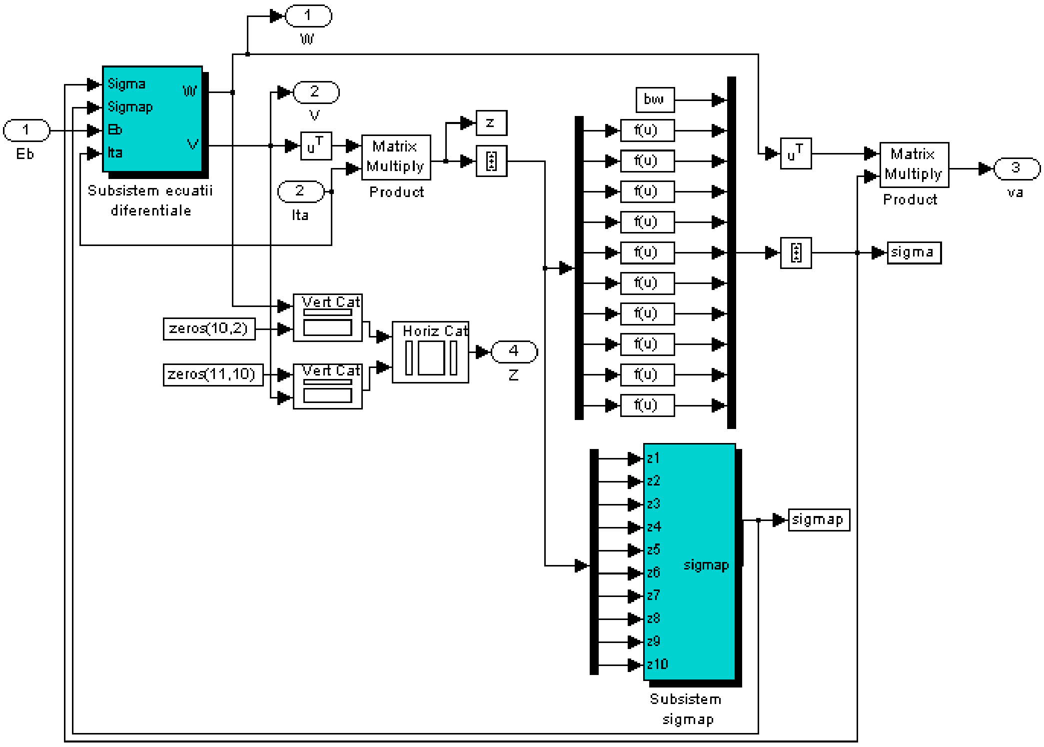
e.


Eb
Ita
Subsistem NNc
Subsistem ita
vc
y
Mat_W
Mat_V
va
vac
Z
Matrix Multiply
Product
Row Sum
sqrt(u)
Zb
kz
Matrix 1-Norm
Ec
P
Bb
kv
vb

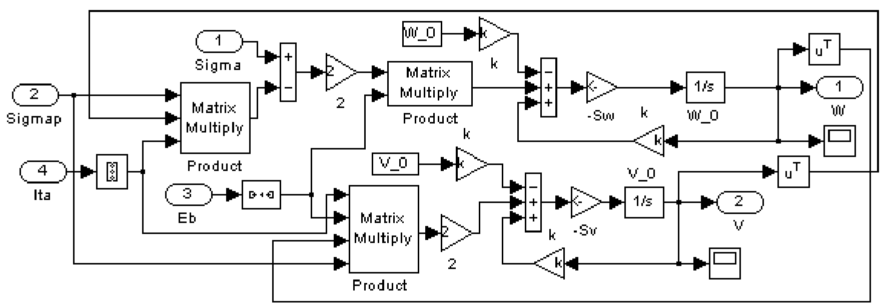
f.


1
vc
2
y
0
zeros(3,1)
zeros(2,1)
Vert Cat
1
Ita

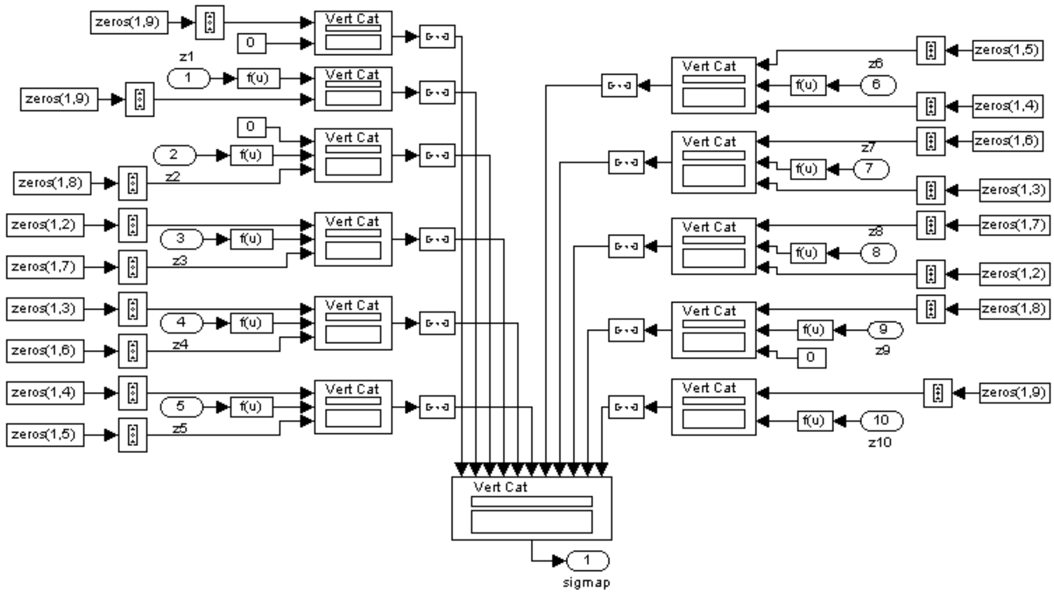
g.


1
Eb
Sigma
Sigmap
Eb
Ita
W
V
Subsistem ecuatii diferentiale
1
W
2
V
2
Ita
Matrix Multiply
Product
z
bw
f(u)
Vert Cat
Horiz Cat
zeros(10,2)
zeros(11,10)
4
Z
Matrix Multiply
Product
3
va
sigma
z1
z2
z3
z4
z5
z6
z7
z8
z9
z10
sigmap
Subsistem sigmap
sigmap

h.


1
Sigma
2
Sigmap
4
Ita
3
Eb
Matrix Multiply
Product
W_0
V_0
k
2
-Sw
-Sv
1/s
1
W
2
V

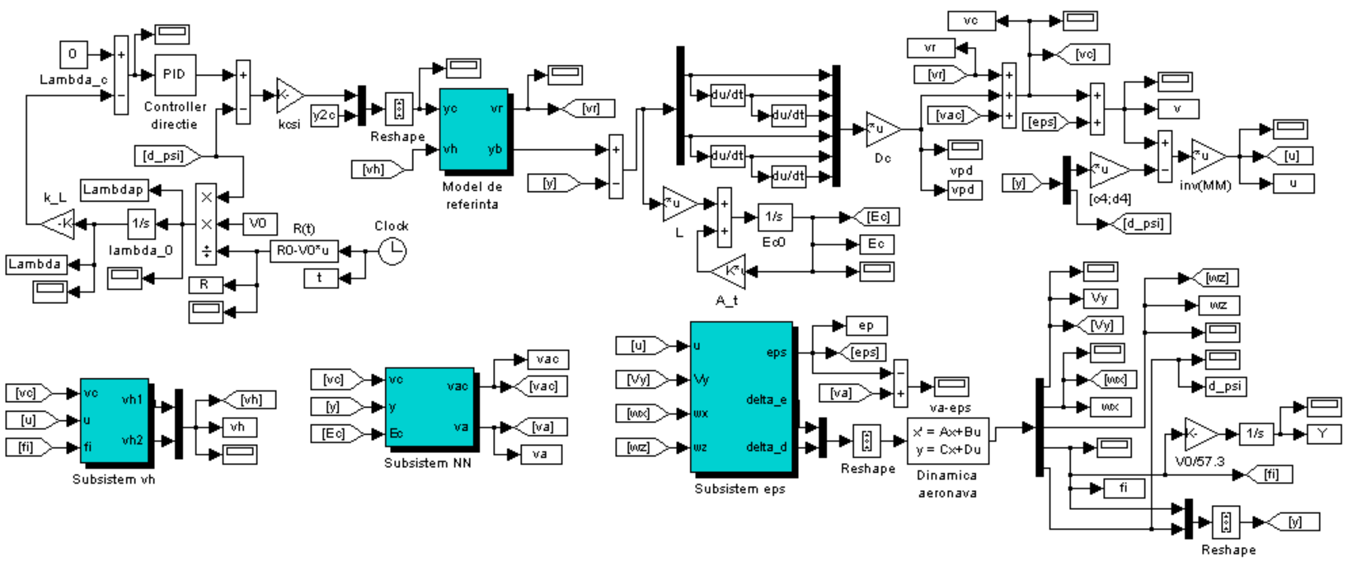
i.



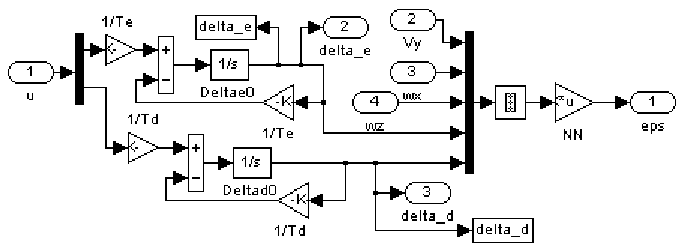
j.

Fig. 3.11. Simulink models for the automatic landing subsystem for aircraft control in longitudinal plane

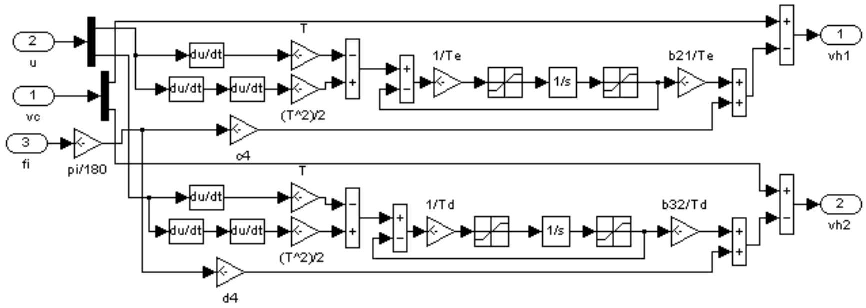
The Matlab/Simulink model for the landing in lateral-directional plane is presented in Fig. 3.12.a. It has for subsystems: b) „reference model (model de referinţă)" (identical with the one from fig. 3.11.b); c) „subsystem eps (subsistem eps)" – fig. 3.12.b; d) „subsystem vh (subsistem vh)" (for the calculation of the output vector $v_h = \begin{bmatrix} v_{h_1} & v_{h_2} \end{bmatrix}^T$ and of the nonlinear model PCH) – fig. 3.12.c; e) „subsystem NN (subsistem NN)" (for the calculation of the components v_a and $\bar{v}$ of the adaptive command $\hat{v}_a$) with the same two subsystems as the subsystem from fig. 3.11.f.



a.



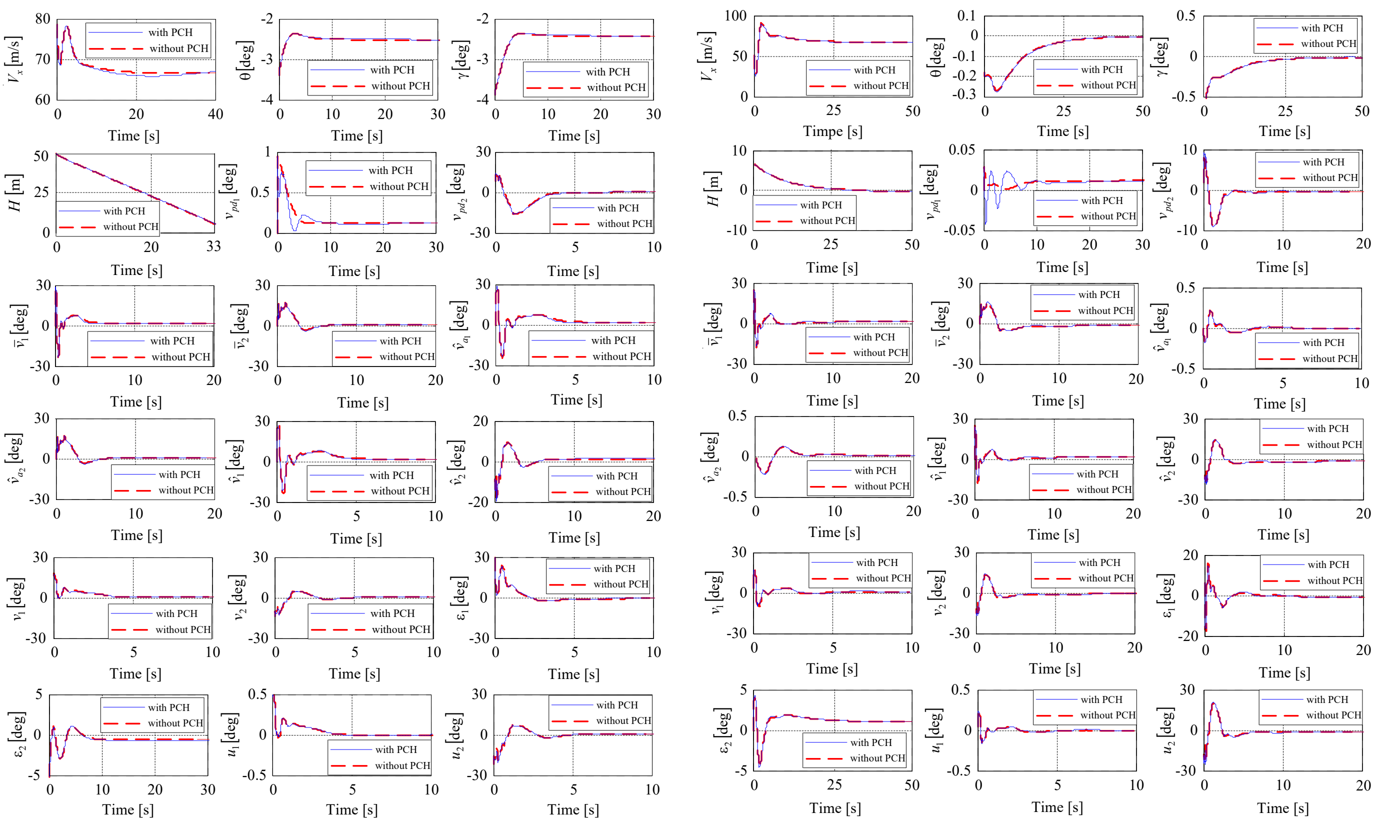
b.



c.

Fig. 3.12. Simulink models of the automatic landing subsystem for aircraft control in lateral-directional plane

The Matlab programs for validation of the complete landing system (longitudinal and lateral-directional planes) are presented in Appendix 5. The new ALS (Fig. 3.10) has been software implemented in Matlab/Simulink and one obtained the time characteristics for the glide slope (Fig. 3.13.a) and flare (Fig. 3.13.b); for lateral plane, one obtained the characteristics in Fig. 3.14. If the actuators are nonlinear, one uses a PCH block $(v_h \neq 0)$ because these allow the system to work in the nonlinearities' linear zones. For the longitudinal plane, the characteristics prove the new ALS's stability and small overshoots. There can be remarked some differences between the case "with PCH block" and "without PCH block". When the actuator is nonlinear, the PCH's usage does not modify the variables' final values, but the transient regime period decreases. One may remark in Fig. 3.13 that the slope angle is in accordance with its theoretical values: -2.5 deg for the glide slope and 0 deg for the flare. During glide slope, the airplane is characterized by a descending trajectory (4th graphic in Fig. 3.13.a), while the flare trajectory is a parabolic one (4th graphic in Fig. 3.13.b) with a null slope angle. Analyzing Fig. 3.13, one remarks that the altitude error is less than 0.5 m during glide slope and 0 m for during flare. These errors meet the Federal Aviation Administration accuracy requirements for best category (Category III); for longitudinal plane, the Category III's accuracy requirements involve altitude error less than 0.5 m (glide slope) and 0 m (flare), while, for lateral-directional plane, lateral deviation Y must be less than 4.1 m. In fig. 3.14, the characteristic $H(X)$ is plotted.



a. Glide slope **b. Flare**

Fig. 3.13. Dynamic characteristics for the new ALS (longitudinal plane)

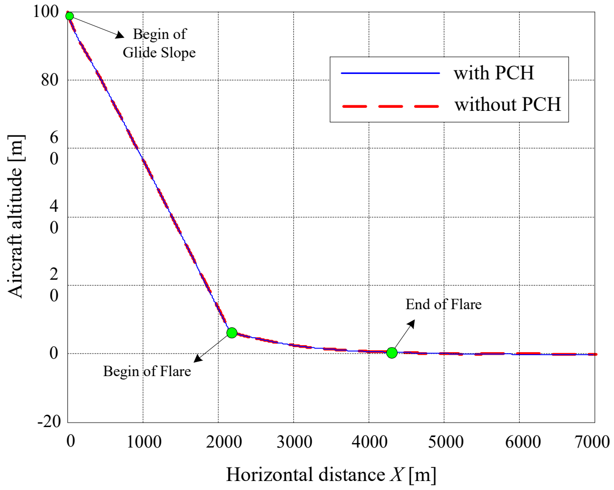


Fig. 3.14. Aircraft altitude versus the horizontal distance

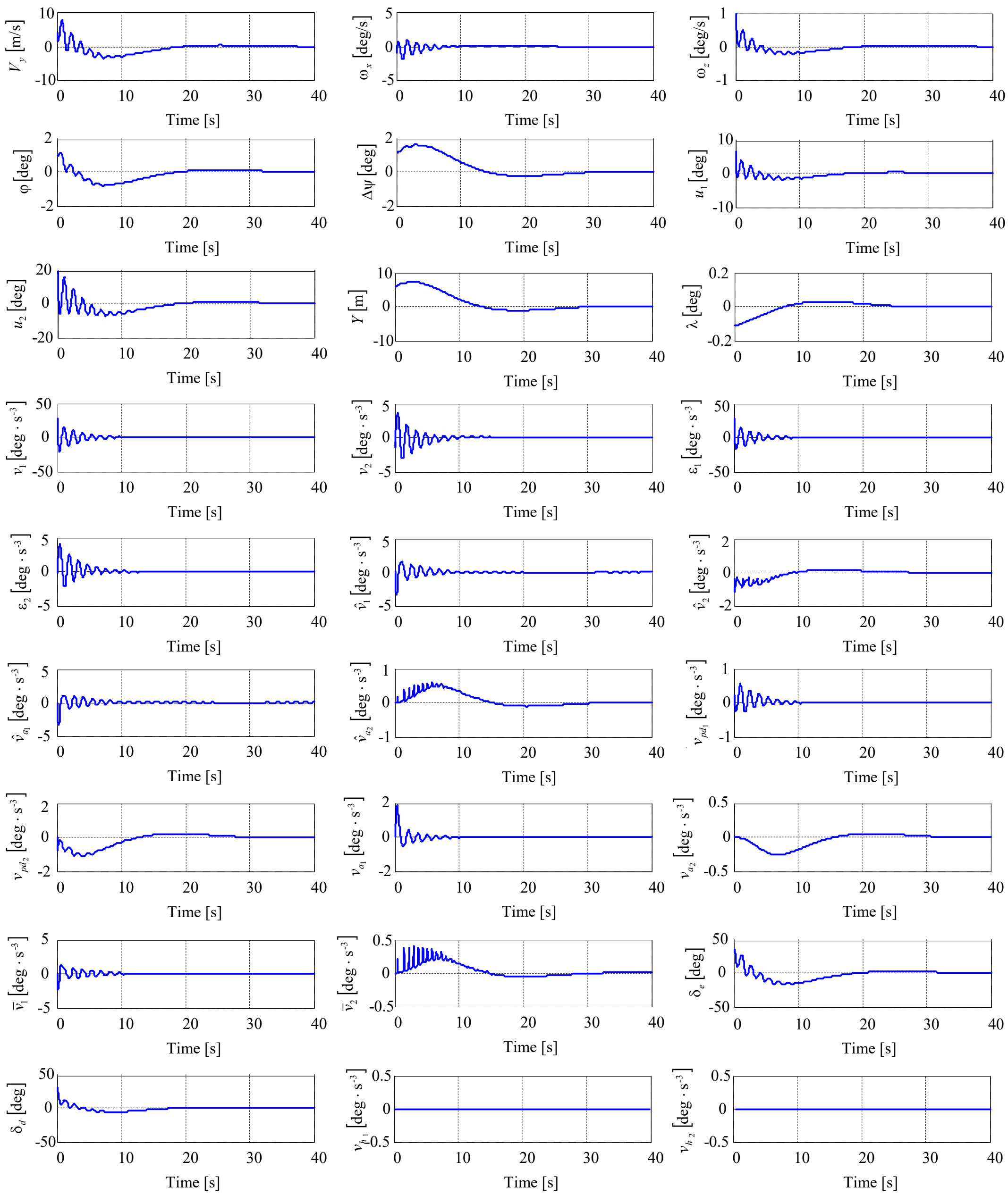


Fig. 3.15. Dynamic characteristics for aircraft motion in lateral-directional plane during landing (case 1: linear actuators)

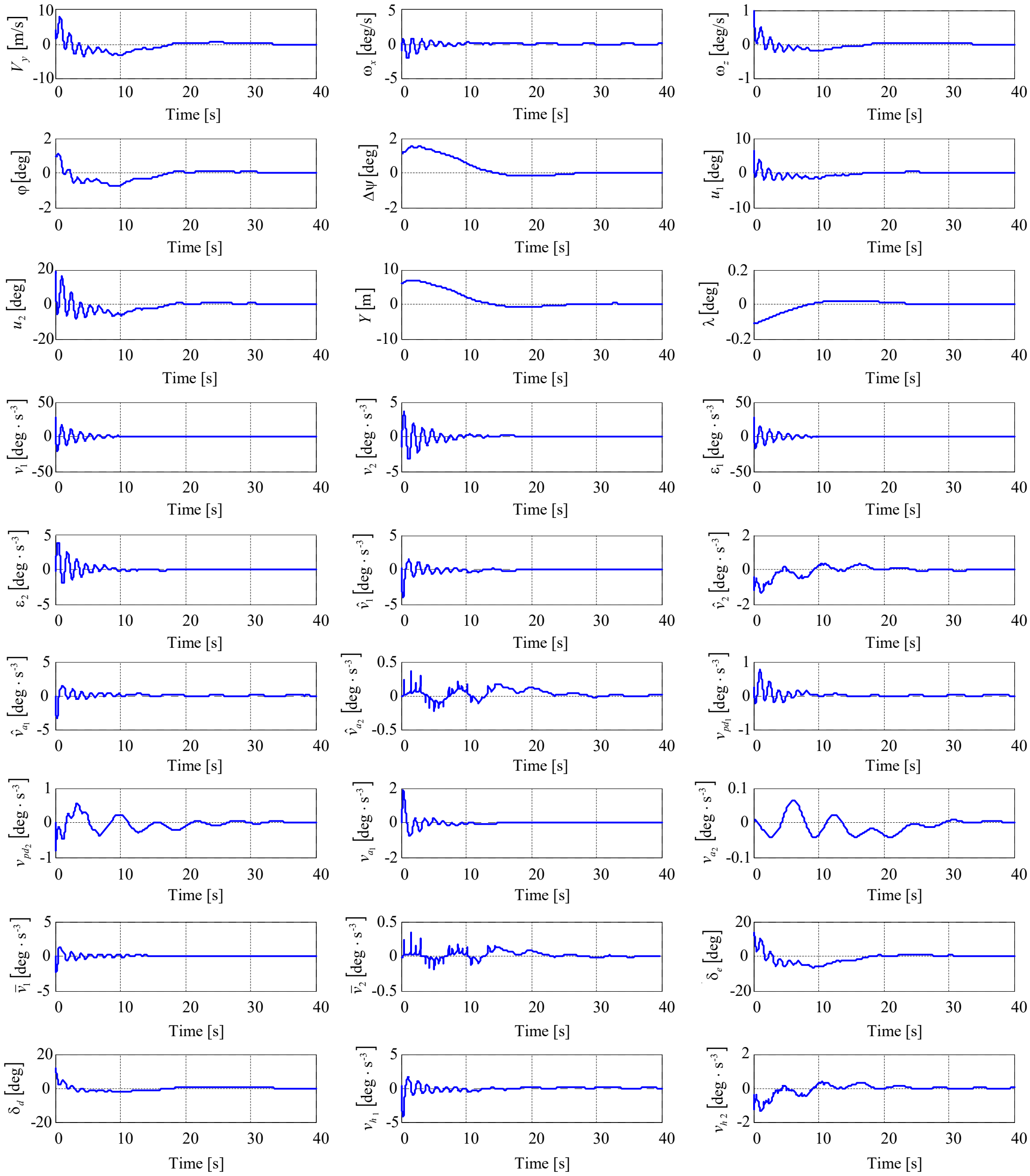


Fig. 3.16. Dynamic characteristics for aircraft motion in lateral-directional plane during landing (case 2: nonlinear actuators)

The glide slope phase takes approximately 33 seconds (see the 7th mini-graphic in fig. 3.13.a), while the flare phase takes approximately 27 seconds (see the 7th mini-graphic in fig. 3.13.b); in the same time, the steady value of aircraft longitudinal velocity is $V_x \cong 67\,\mathrm{m/s}$. Using this information, we make now a brief analysis regarding the correctness of the numerical simulation data: the horizontal distance covered by aircraft in the first landing phase must be approximately $X = 67\,\mathrm{m/s} \cdot 33\,\mathrm{s} = 2211\,\mathrm{m}$, while the horizontal distance covered by aircraft in the second landing phase must be approximately $X = 67\ \mathrm{m/s} \cdot 27\ \mathrm{s} = 1809\ \mathrm{m}$ – values confirmed by fig. 3.14; thus, the total horizontal distance covered during the whole landing process is $X = 2211 + 1809 = 4020\,\mathrm{m}$ – value confirmed again by fig. 3.14.

The new designed architecture for lateral plane is also software implemented in Matlab/Simulink environment; we obtained the time characteristics in fig. 3.15 (case 1: linear servo-actuators) and in fig. 3.16 (case 2: nonlinear servo-actuators). If the actuators are nonlinear, it is good to use a PCH block $(v_h \neq 0)$ because this allows the system to work in the linear zones of the nonlinearities. As we already stated above, the signal provided by the PCH block is a reference model's additional input. Thus, in this paper, we analyze the influence of Pseudo Control Hedging usage on the variables describing the new designed control system for aircraft motion in lateral-directional plane during landing. From figs. 3.15 and 3.16 we remark the cancel of the following variables: the lateral velocity (V_y), the lateral deviation (Y), the roll angle and angular rate (φ, ω_x), the yaw angle and angular rate $(\Delta\psi, \omega_z)$, and the deviation of aircraft longitudinal axis with respect to the runway direction (λ). Also, in steady regime, the command variables ($u_1 = \delta_{e_c}$ and $u_2 = \delta_{d_c}$) as well as the deflections of the ailerons and direction (δ_e and δ_d) cancel. There are also stabilized to zero: the components of the approximation vector ε (ε_1 and ε_2), the components of the pseudo-command $\hat{v}$ ($\hat{v}_1$ and $\hat{v}_2$), the components of the pseudo-command v (v_1 and v_2), the components of the pseudo-command v_{pd} ($v_{pd\,1}$ and $v_{pd\,2}$), the components of the pseudo-command v_a (v_{a1} and v_{a2}), the components of the pseudo-command $\bar{v}$ ($\bar{v}_1$ and $\bar{v}_2$), and the components of the pseudo-command $\hat{v}_a$ ($\hat{v}_{a1}$ and $\hat{v}_{a2}$).

The convergence errors in figs. 3.15 and 3.16 are very good if the Federal Aviation Administration accuracy requirements for Category III (the best category) are analyzed. In our case, the stationary error is very close to zero after 30 seconds; this transient regime is good taking into account that the canceling process of aircraft lateral deviation with respect to the runway takes place long before the start of the two landing main stages in longitudinal plane (the glide slope phase and the flare phase). The reason that the design in this paper meets the requirement and achieves the design goal is that the neural networks have been used.
All the characteristics prove the new architecture's stability and its small overshoots; on the other hand, the differences between the two cases (with and without PCH block) are visible. The usage of a PCH block (when the actuator is nonlinear) does not modify the final values of the variables, but it decrease the transient regime period.

Comparative studies using the two ALSs. Adaptive ALS's optimization and robustness' improvement. Study of the sensor errors and atmospheric disturbances' influence.

The problem of landing has also been discussed in other papers, different types of ALSs being designed. For the same aircraft type, we can make a brief comparison between the obtained characteristics with the ones in [3]; the ALS designed in [3] consists of: an ILS system, a radio-altimeter, and two conventional controllers (a proportional-derivative controller for the pitch angle and a proportional controller for the flight velocity). We can remark that in the case of the adaptive ALS, the dynamic regimes are slightly slower, but the overshoots are smaller, this leading to the conclusion that the neural networks-based adaptive controllers are more efficient than the conventional ones for aircraft landing (see Table 1). The new ALS uses modern design techniques based on the usage of a feed-forward neural network and of the dynamic inversion concept, this bringing advantage over classical control techniques: our new adaptive control methods have applicability to problems involving multivariate systems with cross-coupling between channels; moreover, by using PCH blocks, the non-linear constraints are generally well-handled.

Table 1 – Comparison between the overshoot of the neural networks-based adaptive ALS and conventional one from [3]

Parameters' overshoot	Attack angle [deg]		Pitch angle [deg]		Slope angle [deg]		Velocity V_x [m/s]		Pitch rate [deg/s]	
	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2
ALS with conventional controllers from [3]	2.7	4	4	5	2.8	2.6	19	6	12	11
First neural-network based adaptive ALS	0.5	0.55	1	0.2	1.5	0.5	13	17	1	0.1
Second neural-network based adaptive ALS	0.2	0.16	0.5	0.3	0.5	0.4	4	10	0.2	0.17

For lateral-direction plane, all the characteristics prove the new architecture's stability and its small overshoots; on the other hand, the differences between the two cases (with and without PCH block) are visible. The usage of a PCH block (when the actuator is nonlinear) does not modify the final values of the variables, but it decrease the transient regime period. For the same aircraft type, same direction controller, and radio-navigation system but with a proportional-derivative type control after the roll angle and a proportional type control after $\Delta\psi$ [12], there have been obtained performance inferior to those obtained with the new adaptive system for aircraft landing in lateral-directional plane, both from the transient regime and the overshoots' point of view; also, the results in this paper has been compared to the ones obtained in [2] where the authors have designed two new ALSs by using the H_∞ control, dynamic inversion, optimal observers, and reference models. The time regime period is better in [2] (almost 15 seconds) but the lateral deviation's overshoot is much larger (between 6.15 and 6.32 m depending on the cross wind's velocity) and between

1.10 and 1.31 m for the second designed ALS); therefore, we can conclude that the neural networks-based adaptive controllers are more efficient than the conventional ones for aircraft landing in lateral-directional plane and, using PCH blocks, the nonlinear constraints are better handled.

Influences of disturbances on landing process (longitudinal plane)

We also software validated the two proposed ALSs (one for the longitudinal plane and one for the lateral-directional plane) and, thus, the complete adaptive ALS for the case of Boeing 747's landing affected by different sensor errors and/or wind shears and disturbances.

The linear model of the aircraft motion, in longitudinal plane, can e described by the state equation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{B}_{\mathbf{v}}\mathbf{v}_{\mathbf{v}}, \tag{3.18}$$

with $\mathbf{x}$- the state vector, $\mathbf{u}-$ the command vector (2×1), $\mathbf{v}_{\mathbf{v}}-$ the vector of disturbances V_{vx} and V_{vz} (the components of the wind velocity along aircraft axes Ox and Oz),

$$\mathbf{x} = \begin{bmatrix} V_x & \alpha & \omega_y & \theta \end{bmatrix}^T, \mathbf{u} = \begin{bmatrix} \delta_p & \delta_T \end{bmatrix}^T, \mathbf{v}_{\mathbf{v}} = \begin{bmatrix} V_{vx} & V_{vz} \end{bmatrix}^T; \tag{3.19}$$

in above equations ω_y is the pitch angular rate, δ_p- the elevator deflection, δ_T- the engine command, while the matrices $\mathbf{A}, \mathbf{B}, \mathbf{B}_{\mathbf{v}}$ have the forms:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g\cos\theta_0 \\ Z_u^* & Z_w & 1 & -(g/V_0)\sin\theta_0 \\ \widetilde{N}_u & \widetilde{N}_w & \widetilde{N}_q & \widetilde{N}_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{21} & b_{31} & 0 \\ b_{12} & b_{22} & b_{32} & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & Z_{\delta_p}/V_0 & \widetilde{N}_{\delta_p} & 0 \\ X_{\delta_T} & Z_{\delta_T}/V_0 & N_{\delta_T} & 0 \end{bmatrix}^T,$$
$$\mathbf{B}_{\mathbf{v}} = \begin{bmatrix} -a_{11} & -a_{21} & -a_{31} & 0 \\ -a'_{12} & -a'_{22} & -a'_{32} & 0 \end{bmatrix}^T = \begin{bmatrix} -a_{11} & -a_{21} & -a_{31} & 0 \\ -a_{12}/(57.3V_0) & -a_{22}/(57.3V_0) & -a_{32}/(57.3V_0) & 0 \end{bmatrix}^T; \tag{3.20}$$

the elements of the matrices $\mathbf{A}, \mathbf{B}, \mathbf{B}_{\mathbf{v}}$ have been calculated with respect to the stability derivates for the chosen aircraft type. For the modeling of the wind shears we use the equations in [5], [13]:

$$V_{vx} = -V_{vx0}\sin(\omega_0 t),\ V_{vz} = -V_{vz0}\left[1-\cos(\omega_0 t)\right], \omega_0 = 2\pi/T_0, \tag{3.21}$$

where T_0 is the flight time period inside the wind shear; the aircraft faces head wind and rear wind combined with vertical wind. A deterministic model of the atmospheric turbulences have been proposed by Karman & Dryden and adopted by the International Civil Aviation Organization Standard Atmosphere, while a spectral model of the atmospheric turbulences can be also found in literature. We used the Dryden spectral model described by equations:

$$V_{vx} = N_1\sqrt{\frac{1}{\Delta t}}\frac{\sigma_u\sqrt{2\frac{L_u}{V_0}}}{1+\frac{L_u}{V_0}\mathrm{s}}, V_{vz} = N_1\sqrt{\frac{1}{\Delta t}}\sigma_w\sqrt{\frac{L_w}{\pi V_0}}\frac{1+\frac{\sqrt{3}L_w}{V_0}\mathrm{s}}{\left(1+\frac{L_w}{V_0}\mathrm{s}\right)^2}, \tag{3.22}$$

where

$$L_u = \frac{H}{(0.177+0.000832H)^{1.2}}, L_w = H,\ \sigma_u = \frac{\sigma_w}{(0.177+0.000832H)^{0.4}}, \sigma_w = 0.1\cdot V_{20}, V_{20} = 20\,\text{ft/s}; \tag{3.23}$$

H is expressed in ft. In equation (3.23), Δt is the simulation step, N_1- the Gaussian white noise with zero mean and unity standards in deviation, while $V_0[\text{ft/s}]$ is the aircraft nominal velocity [13].

In Figs. 3.17 and 3.18 we represent the time characteristics for the glide slope landing phase and flare landing phase, respectively. The characteristics have been represented in the presence or in the absence of the wind shears which are modeled by equations (3.21) with $T_0 = 60\,\text{s}$, $V_{vx_0} = 10\,\text{m/s}$, $V_{vz_0} = 15\,\text{m/s}$. To be sure that the new ALS responds very well to wind shears, we have chosen values of V_{vx_0} and V_{vz_0} that are bigger than their medium values [13]. The presence of the wind shears is not very visible – the curves with solid line (without wind shears) overlap almost perfectly over the curves plotted with dashed line (with wind shears). The time origin for the flare trajectory is chosen zero when the altitude is $H = H_0 = 20\,\text{m}$ (the altitude at which the glide slope process ends).

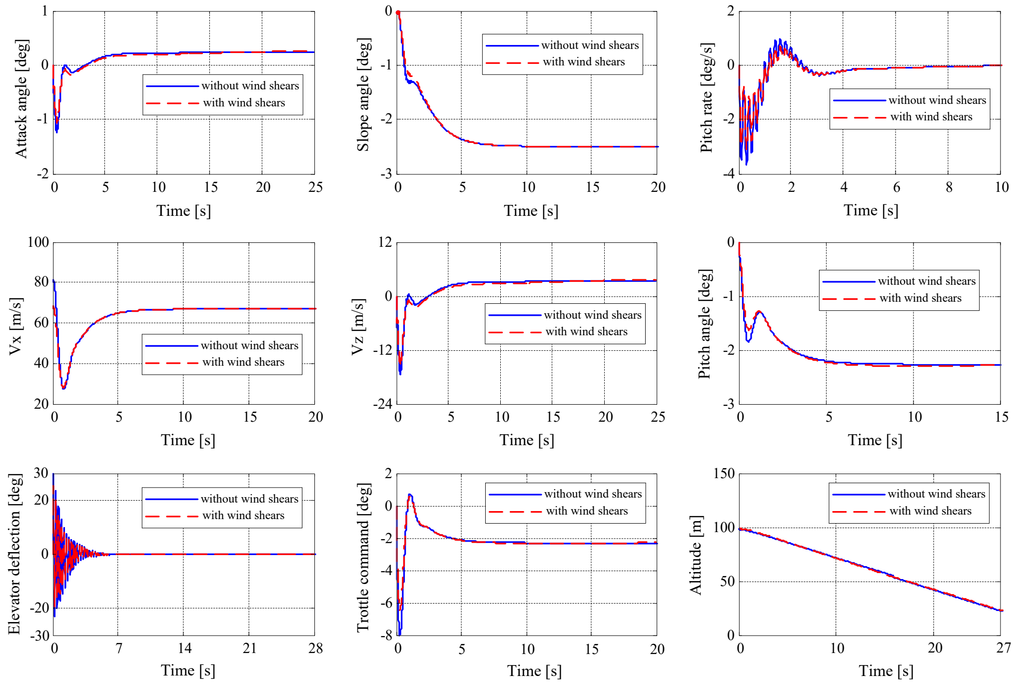


Fig. 3.17. Time characteristics of the adaptive ALS (glide slope phase), with or without wind shears

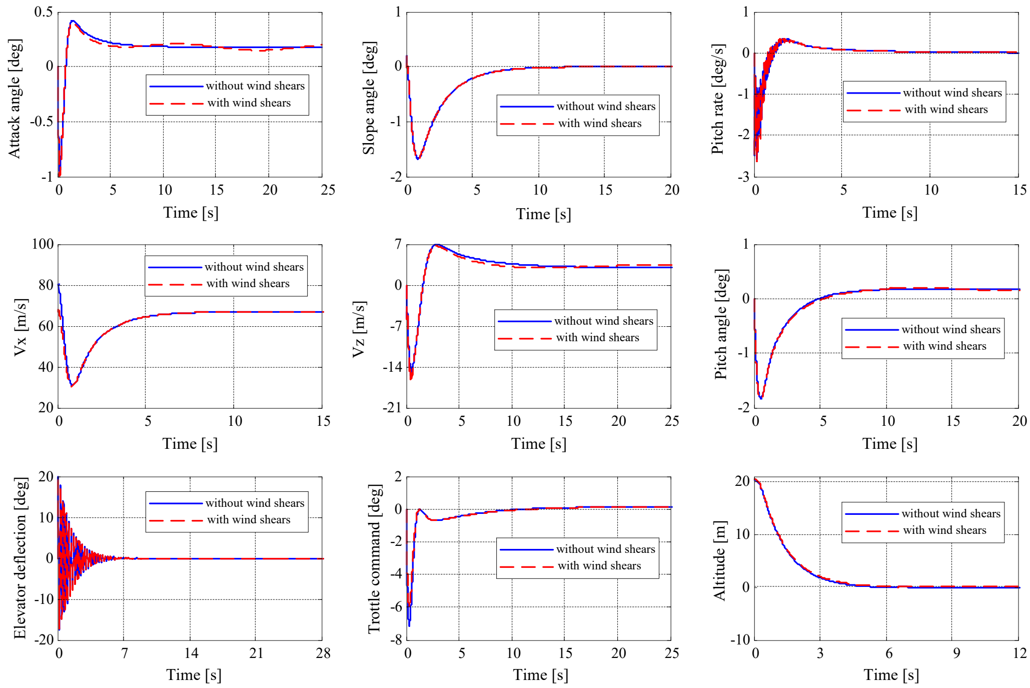


Fig. 3.18. Time characteristics of the adaptive ALS (flare), with or without wind shears

In Figs. 3.19 and 3.20 we present the time characteristics of the adaptive ALS (longitudinal plane), for the glide slope landing phase and flare landing phase, respectively, taking into account or not the atmospheric turbulences described by equations (3.22) and (3.23) – the Dryden spectral model [13]; in Fig. 3.21 the velocities $V_{vx}(t)$ and $V_{vz}(t)$, modeled by the Dryden equations (3.22) and (3.23), are represented and were generated by using an already built Matlab model from the “Aerospace Blockset” toolbox.

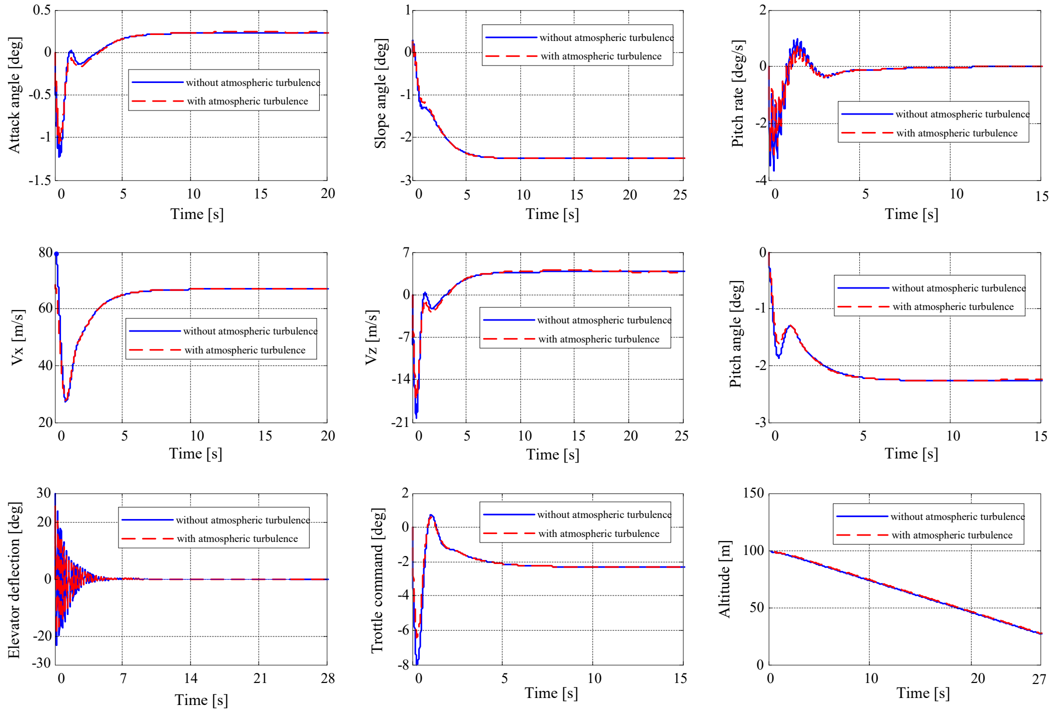


Fig. 3.19. Time characteristics of the adaptive ALS (glide slope phase), taking into account or not the turbulences

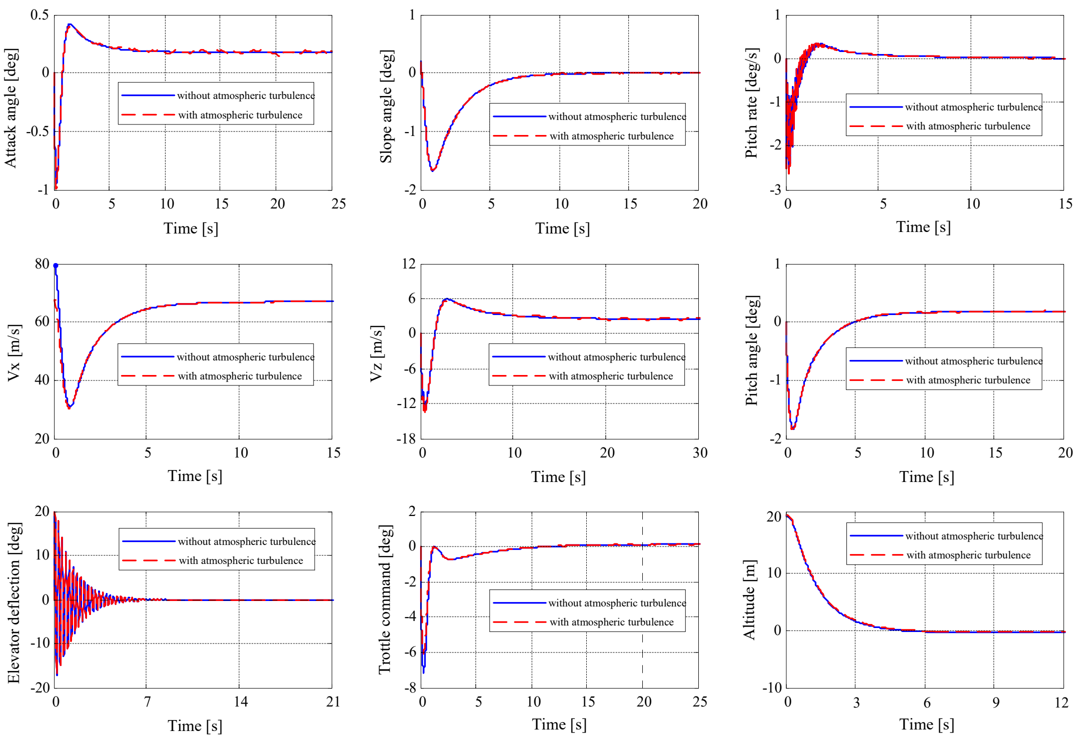


Fig. 3.20. Time characteristics of the adaptive ALS (flare), taking into account or not the turbulences

In the above simulations we did not take into consideration the errors of the sensors (the sensors are used for the measurement of the states). The errors of the gyro sensors in the pitch channel are considered in simulations below. For the determination of the pitch angle θ we may use an integrator gyro [13]. This gyro has errors and it is interesting to see if the sensor errors affect the landing process; its error model takes into account the parameters from the data sheets offered by the producers of the sensors. The error model is described by the equation: $\theta = (\theta_i + S \cdot a_r + B + \nu)\left(1 + \frac{\Delta K}{K}\right)$, where θ is the output pitch angle (the perturbed signal), θ_i – the input pitch angle, S – the sensibility to the acceleration a_r applied on an arbitrary direction, B – the bias, K – the scale factor, ΔK – the calibration error of the scale factor, and ν – the sensor noise [13].

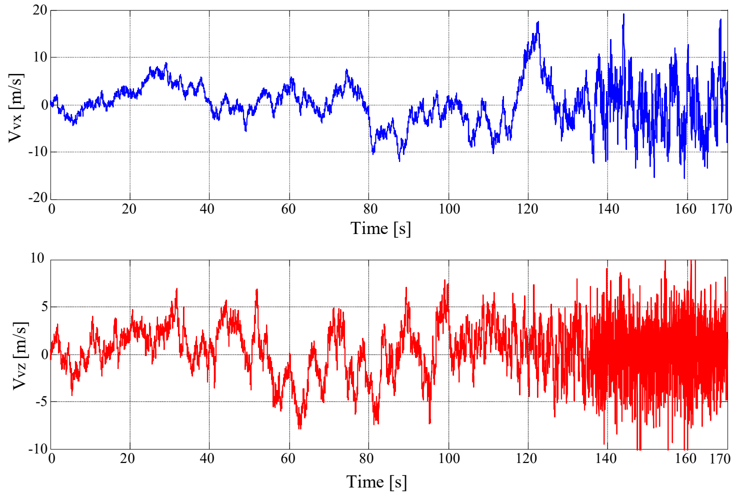


Fig. 3.21. The velocities $V_{vx}(t)$ and $V_{vz}(t)$ modeled by the Dryden equations (3.22) and (3.23)

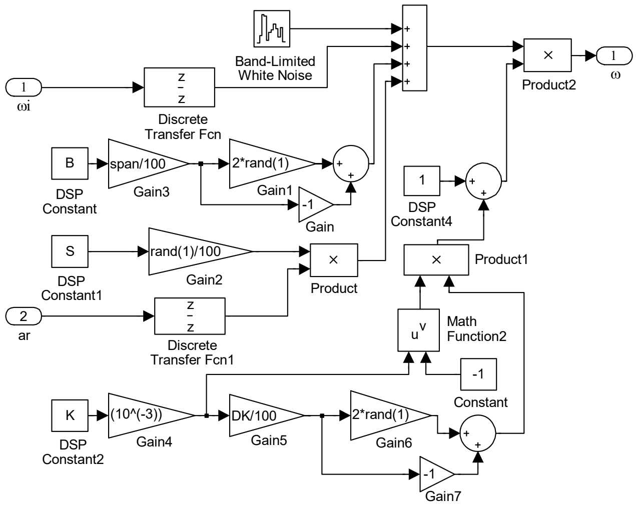


Fig. 3.22. The error model for the gyro in Matlab/Simulink

A Matlab/Simulink model (Fig. 3.22) has been introduced in the model from Fig. 3.11 in the feedback after the pitch angle θ. In the model, the bias is given by its maximum value B as percentage of span, the calibration error of the scale factor is given by its absolute maximum value ΔK as percentage of K, while the noise is given by using its maximum density value. Using the Matlab function "rand(1)" we generate the bias, by a random value in the interval $(-B, B)$, the sensibility S to acceleration a_r, applied on an arbitrary direction in the interval $(0, S)$, and the calibration error of the scale factor in the interval $(-\Delta K, \Delta K)$. The noise is generated by means of a Simulink block "Band-Limited White Noise" by using the Matlab function "RandSeed" which generates a random value of its density in the interval $(80\% \cdot \nu_d, \nu_d)$. The inputs of the error model are the pitch angle θ_i and the acceleration a_r, considered to be the resultant acceleration signal that acts upon the carrier vehicle, while the output is the disturbed pitch angle θ. In the numerical simulation, the following sensor parameters have been used: the noise density: $5.8 \cdot 10^{-6}\left[\mathrm{deg}/\sqrt{\mathrm{Hz}}\right]$, the bias: $9.8 \cdot 10^{-8}[\mathrm{deg}]$, the error of the scale factor: $5.2 \cdot 10^{-4}\% \cdot K$, the sensibility to accelerations $\cong 0[\mathrm{deg}/g]$; g is the gravitational acceleration [13].

Figs. 3.23 and 3.24 depict the time characteristics of the adaptive ALS, for the glide slope phase and flare phase, respectively, taking into account or not the errors of the gyro sensor. Although the errors of the gyro sensor affect some variables, the time variation of the altitude (the last mini-graphic) and the time length of the landing process phases are not affected. So, the authors may conclude that the sensor errors do not affect the landing process [13].

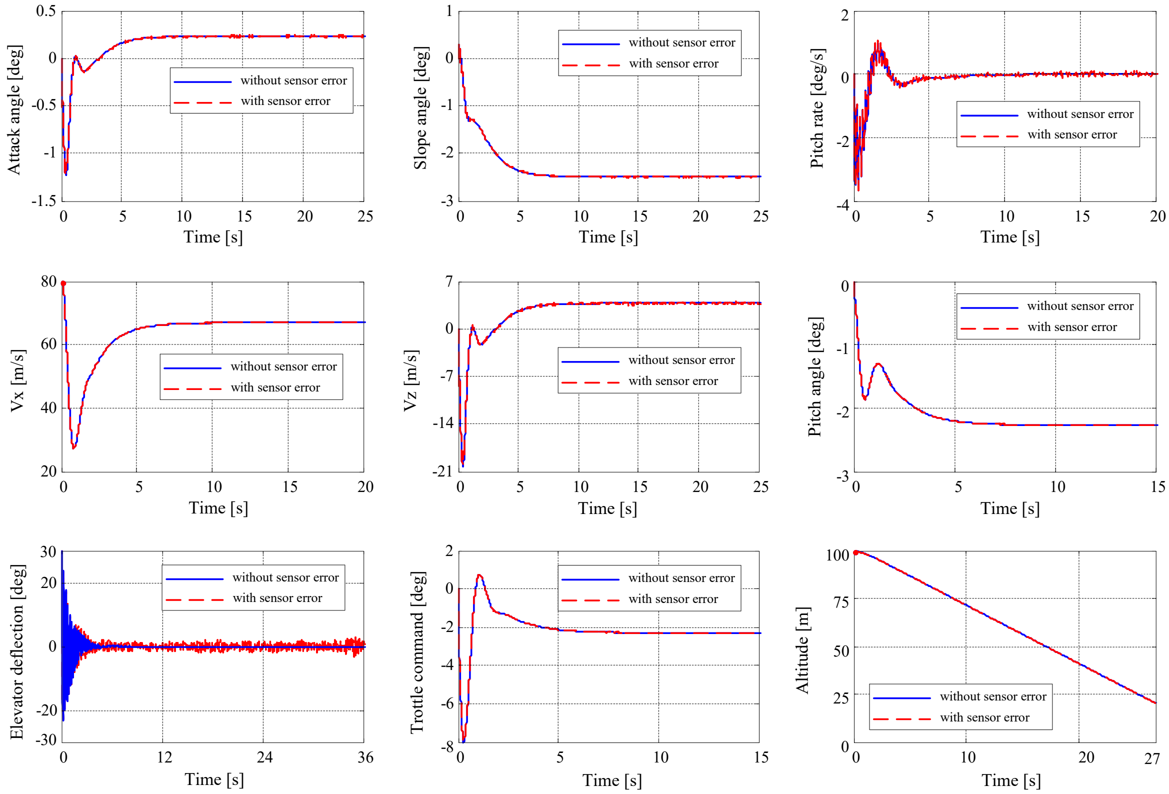


Fig. 3.23. Time characteristics of the adaptive ALS (glide slope phase), taking into account or not the sensor errors

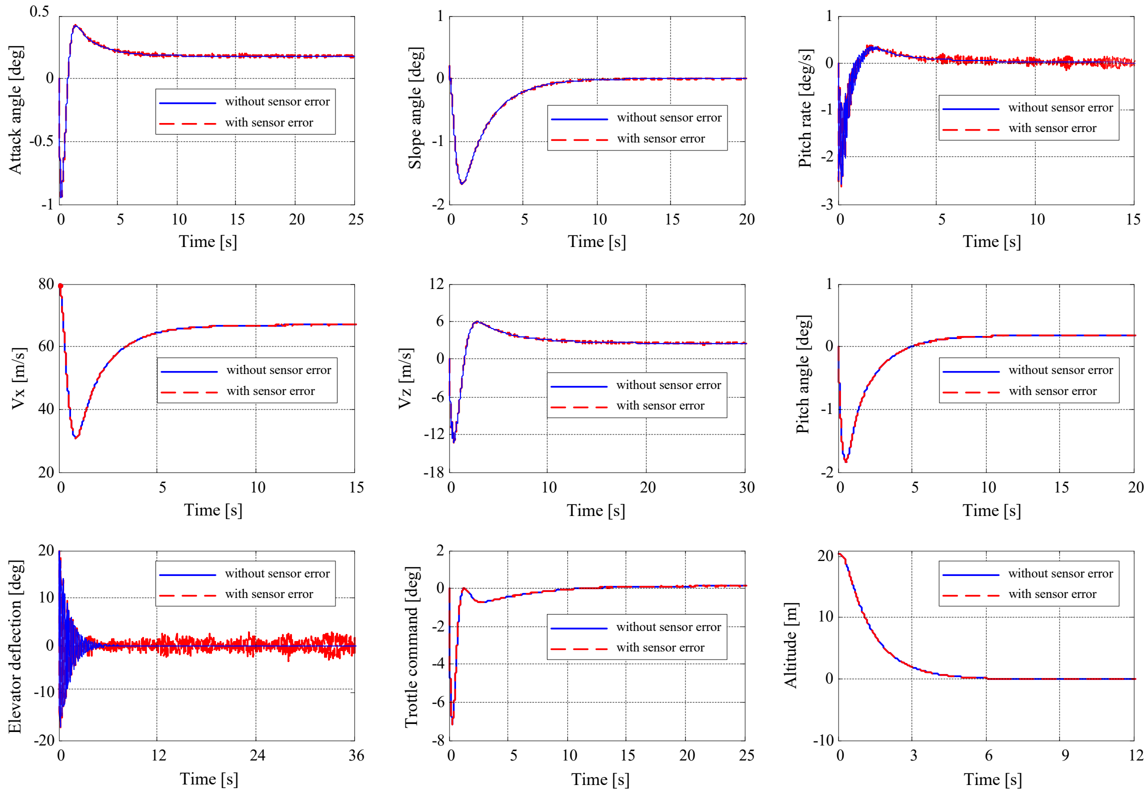


Fig. 3.24. Time characteristics of ALS with PID controllers, for flare phase, taking into account or not the sensor errors

In Table 2 we present, for the most important 5 variables, the influences of the wind shears, atmospheric turbulences, and errors of the sensor upon the parameters maximum absolute deviation with respect to their steady values. From the values in Table 2 it can be noticed that the atmospheric turbulences have the most significant influence on the steady – state values of the variables, while the sensor errors have the less significant influence. The parameters maximum absolute deviation is almost the same for the first two considered disturbances: for example, in the attack angle case (Phase 2 – Flare), and in the pitch angle case for both landing phases; generally, the ratio of the errors induced by the turbulences and sensor errors is 3:1 [13].

Table 2 – Parameters maximum absolute deviation with respect to their steady values (ALS with PID Controllers)

Parameters' maximum absolute deviation with respect to their steady values	Attack angle α[deg]		Pitch angle θ[deg]		Slope angle γ[deg]		Velocity V_x [m/s]		Velocity V_z [m/s]	
	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2
ALS affected by wind shears	0.0235	0.0314	0.031	0.0298	0.0045	0.00469	0.0162	0.0228	0.4416	0.4356
ALS affected by turbulences	0.0447	0.0385	0.0352	0.0242	0.0136	0.0210	0.0555	0.090	0.5412	0.6168
ALS affected by sensor errors	0.0124	0.0141	0.0041	0.0061	0.0136	0.0119	0.0722	0.077	0.1764	0.2388

Influences of disturbances on landing process (lateral-directional plane)

We do the same thing for the landing process in lateral-directional plane; here, the influences of wind shears and disturbances are not so important, but the influence of sensor errors should be analyzed. In Fig. 3.25 we represent the time characteristics for the flight direction control subsystem; before the start of the two landing main stages in longitudinal plane, the pilot must cancel the aircraft lateral deviation with respect to the runway. The characteristics have been represented for the ALS affected by crosswind in the presence or in the absence of sensor errors (the sensors are used for the measurement of the states). The presence of the sensor errors is not visible – the curves with solid line (obtained for the ALS without sensor errors) overlap almost perfectly over the curves plotted with dashed line (obtained for the ALS with sensor errors).

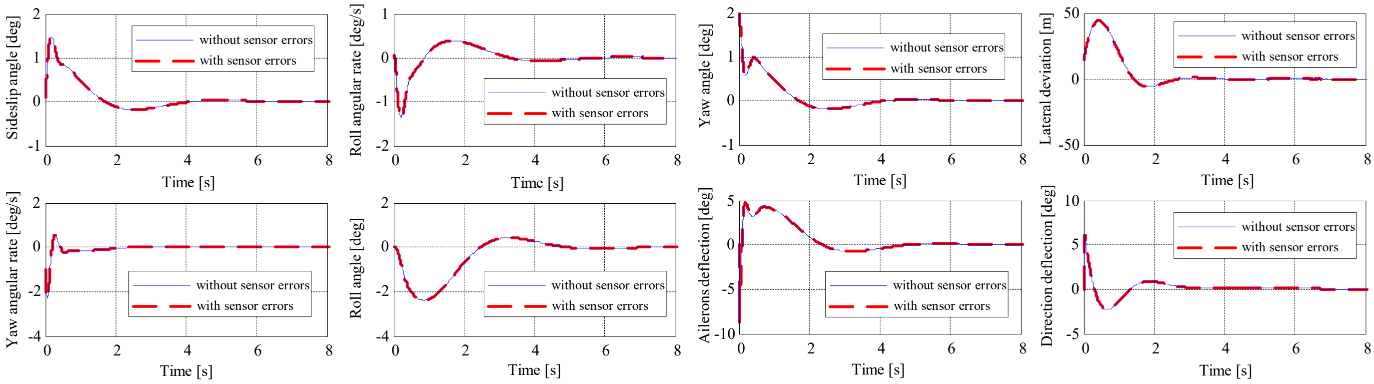


Fig. 3.25. Time characteristics of the lateral-directional adaptive ALS, with or without sensor errors

The landing approach begins at 67 m/s nominal speed; this should be maintained constant. To test the robustness of the designed ALS, in simulations, we have taken into consideration the crosswind, because low-altitude crosswind can be a serious threat to the safety of aircraft in landing. From the 4th mini-graphic in Fig. 3.25, we can see that the stationary value of the aircraft lateral deviation is very close to zero; this error is very good if we analyze the Federal Aviation Administration accuracy requirements for Category III (best category). If the crosswind is stronger than its maximum accepted value, the pilot must avoid having the aircraft enter into this wind shear. From the sideslip angle point of view, the errors are less than 0.1 deg.; we conclude that $\beta \rightarrow \beta c=0$ deg.

Activity III.9. Results' dissemination

The 9th activity of project's second stage (2017) was intended to carry out the objective ***OS21*** (*To disseminate the results in the scientific, academic and socio-economic environment*). It was conducted throughout the stage and materialized through a web page actualization with all the information regarding the achievements during Stage II and the accomplishment of the project's objectives. The links for the web page are:

- http://elth.ucv.ro/site/mlungu/index.php?language=ro&page=48 (Romanian);
- http://elth.ucv.ro/site/mlungu/index.php?language=en&page=50 (English)

From the expected results' point of view, the research team had the following targets during 2017:

5) *Design of a new adaptive control system using the dynamic inversion and neural networks*. **This aim of the project has been acomplished by designing the automatic landing system from Fig. 3.10**.
6) *Software package for the new automatic landing system*. This aim of the project has been acomplished by obtaining the **Matlab/Simulink programs from Appendix 5 (landing in two planes: longitudinal and lateral) and the Simulink models from Figs. 3.11 and 3.12**.
7) *Phase repport* (**the present document**).
8) *Publishing of 1 paper in ISI Journals and 6 papers in other databases*. In 2017, the members of the research team have published **3 papers in ISI Journals and 5 conference papers in international databases (ISI Web of Science, IEEE Xplore, Google Scholar, Engineering & Technology Digital Library, Crossref, DOAJ etc.)**. Thus, **the target related to the papers' publishing has been reached and even exceeded**. Below, we present the 8 papers (with a short description) published in 2017 by the members of the research team.
   - 4.1. Lungu, R., Lungu, M. *Automatic control of the micro aerial vehicles' attitude and position*. International Journal of Micro Aerial Vehicles, vol. 9, nr. 1, pag. 61-73, 2017, ISSN: 1756-8293 (ISI Journal). *Databases: ISI Web of Science*.
   - 4.2. Lungu, R., Lungu, M. *Automatic Landing System using Neural Networks and Radio-technical Subsystems*. Chinese Journal of Aeronautics, vol. 30, nr. 1, pag. 399-411, 2017, ISSN: 1000-9361 (ISI Journal). *Databases: ISI Web of Science*.
   - 4.3. Lungu, M., Lungu, R. *Landing Auto-pilots for Aircraft Motion in Longitudinal Plane using Adaptive Control Laws Based on Neural Networks and Dynamic Inversion*. Asian Journal of Control, vol. 19, nr. 1, pag. 302-315, 2017, ISSN**:** 1561-8625 (ISI Journal). *Databases: ISI Web of Science*.
   - 4.4. Lungu, M., Lungu, R. *The Control of Airplane Landing in Longitudinal and Lateral-directional Planes by using the H-inf Control*. 18^{th} International Carpathian Control Conference, Sinaia, Romania, 28-31 May 2017 (ISI Proceedings). *Databases*: *ISI Web of Science, IEEE Xplore*.
   - 4.5. Lungu, M., Lungu, R. *Complete Landing Autopilot having Control Laws Based on Neural Networks and Dynamic Inversion*. 18^{th} International Carpathian Control Conference, Sinaia, Romania, 28-31 May 2017 (ISI Proceedings). *Databases*: *ISI Web of Science, IEEE Xplore*.
   - 4.6. Lungu, M. *Automatic control of aircraft landing by using the H_2/H_∞ control technique*. The 36^{th} IASTED International Conference on Modelling, Identification and Control (MIC 2017), February 20-21, 2017, Innsbruck, Austria, DOI: 10.2316/P.2017.848-004, pp. 222-229. *Database*: *Scopus*.
   - 4.7. Lungu, M., Lungu, R. *Landing Autopilot for the Control of Airplane by using the H-inf Control*. 12^{th} Int. Conference on Aerospace, Robotics, Mechatronics, Mechanical Engineering, Manufacturing systems, Neurorehabilitation and Bioengineering (OPTIROB), 29 Iunie-3 Iulie 2017, Jupiter. Int. Journal of Modeling and Optimization (IJMO), vol. 7, nr. 3, 2017, pag. 163-167. *Databases: Google Scholar, EI (INSPEC, IET), Engineering & Technology Digital Library, Crossref, DOAJ.*
   - 4.8. Voicu, S., Buţu, F. *H-Infinity Design for Automatic Landing System*. 12^{th} Int. Conference on Aerospace, Robotics, Mechatronics, Mechanical Engineering, Manufacturing systems, Neurorehabilitation and Bioengineering (OPTIROB), June 29- July 3, 2017, Jupiter. Int. Journal of Modeling and Optimization (IJMO), vol. 7, nr. 3, 2017. *Databases: Google Scholar, EI (INSPEC, IET), Engineering & Technology Digital Library, Crossref, DOAJ.*

Short description of the 8 published papers:

- Lungu, R., Lungu, M. *Automatic control of the micro aerial vehicles' attitude and position*. International Journal of Micro Aerial Vehicles, vol. 9, nr. 1, pag. 61-73, 2017, ISSN: 1756-8293 (ISI Journal). *Databases: ISI Web of Science*.

The paper focuses on two automatic systems for the attitude and position's control of the micro aerial vehicles (MAVs) – insect type by using a nonlinear dynamic model which describes the motion of MAVs with respect to the Earth tied frame. The attitude control is adaptive type, with the estimation of the inertia moments' matrix and of the dynamic damping coefficients' matrix in two variants: by means of the attitude vector or by using the quaternion vector. The new resulting control architectures use a vector for the control of the MAV's attitude, a proportional-derivative linear dynamic compensator, an error vector (whose elements are the estimated deviations of the inertia moments and dynamic damping coefficients with respect to the real ones), and the Lyapunov theory. In the two variants of the adaptive control, the control law is represented by the command aerodynamic moments and the wing rotation's command vector, respectively; the control law for the MAV position's control is deduced in the same way. The two obtained control systems are validated by complex numerical simulations.

- Lungu, R., Lungu, M. *Automatic Landing System using Neural Networks and Radio-technical Subsystems*. Chinese Journal of Aeronautics, vol. 30, nr. 1, pag. 399-411, 2017, ISSN: 1000-9361 (ISI Journal). *Databases: ISI Web of Science*.

The paper focuses on the design of a new Automatic Landing System (ALS) in longitudinal plane; the new ALS controls the aircraft trajectory and longitudinal velocity. Aircraft control is achieved by means of a proportional-integral (P.I.) controller and the Instrumental Landing System – first phase of landing (the glide slope) and a proportional-integral-derivative (P.I.D.) controller together with a radio-altimeter – second phase of landing (the flare), respectively; both controllers modifies the reference model associated to aircraft pitch angle. The control of the pitch angle and of the longitudinal velocity is performed by a neural network adaptive control system, based on the dynamic inversion concept, having as components: a linear dynamic compensator, a linear observer, reference models, and a Pseudo Control Hedging (PCH) block. The theoretical results are software implemented and validated by complex numerical simulations; compared with other ALSs having the same radio-technical subsystems but with conventional or fuzzy controllers for the control of aircraft pitch angle and longitudinal velocity, the architecture designed in this paper is characterized by much smaller overshoots and stationary errors.

- Lungu, M., Lungu, R. *Landing Auto-pilots for Aircraft Motion in Longitudinal Plane using Adaptive Control Laws Based on Neural Networks and Dynamic Inversion*. Asian Journal of Control, vol. 19, nr. 1, pag. 302-315, 2017, ISSN**:** 1561-8625 (ISI Journal). *Databases: ISI Web of Science*.

The paper presents two new automatic landing systems (ALSs) for aircraft motion in longitudinal plane; the model of the landing geometry determines the flight trajectory and the aircraft calculated altitude; the flight trajectory during landing consists of two parts: the glide slope and the flare. Both designed ALSs have an adaptive system (ACS) for the aircraft output's control; for the first ALS, the output vector consists of the flying altitude and the longitudinal velocity, while, for the second ALS, the output variables are the pitch angle and the longitudinal velocity of aircraft. The second variant of ALS also contains an altitude controller providing the calculated pitch angle. The calculated altitude (for the first ALS), the calculated pitch angle (for the second ALS), and the desired flight velocity are provided to the ACS by means of a block consisting of two reference models. ACS is based on the dynamic inversion concept and contains an adaptive controller which includes a linear dynamic compensator, a state observer, a neural network, and a Pseudo Control Hedging (PCH) block. The paper is focused both on the design of the two ALSs and on their complex software implementation and validation.

- Lungu, M., Lungu, R. *The Control of Airplane Landing in Longitudinal and Lateral-directional Planes by using the H-inf Control*. 18th Int. Carpathian Control Conference, Sinaia, Romania, 28-31 May 2017 (ISI Proceedings). *Databases: ISI Web of Science, IEEE Xplore*.

This work presents the automatic control of airplane in longitudinal and lateral-directional planes, during landing, by using the H-inf control and the airplane's linearized dynamics; the H-inf control technique assures robust stability with respect to different disturbances and noise type signals. Both the longitudinal plane and the lateral-directional one are treated; three automatic landing subsystems (ALSs) are obtained: the first two control airplane's motion in longitudinal plane, while the third one is employed for the airplane's control in lateral-directional plane. Before the landing's two main stages in longitudinal plane, the pilot must cancel the airplane's lateral deviation with respect to the runway in lateral plane. The three subsystems have been designed separately and put together to form a single ALS. The complete ALS has been software implemented, tested, and validated through numerical simulations; promising results have been obtained.

- Lungu, M., Lungu, R. *Complete Landing Autopilot having Control Laws Based on Neural Networks and Dynamic Inversion*. 18th Int. Carpathian Control Conference, Sinaia, Romania, 28-31 May 2017 (ISI Proceedings). *Databases: ISI*

Web of Science, IEEE Xplore.

This paper discusses the automatic control of airplane in longitudinal and lateral-directional planes, during landing, using two neural network and dynamic inversion concept based adaptive control subsystems, having as components: linear observers, linear dynamic compensators, reference models, and Pseudo Control Hedging blocks in order to eliminate the adapting difficulties of the neural network appearing because of the actuators' nonlinearities. Two subsystems (for longitudinal and lateral-directional planes) will be put together and a new complete landing auto-pilot will result; it is software implemented the results' accuracy and the achievement of the precision standards being briefly analyzed.

- Lungu, M. *Automatic control of aircraft landing by using the H_2/H_∞ control technique*. The 36th IASTED International Conference on Modelling, Identification and Control (MIC 2017), February 20-21, 2017, Innsbruck, Austria, DOI: 10.2316/P.2017.848-004, pp. 222-229. *Database: Scopus*.

The paper presents the automatic control of aircraft in longitudinal plane, during landing, by using the linearized dynamics of aircraft, taking into consideration the sensor errors and other external disturbances. Aircraft auto-landing is achieved by combining two control techniques: H_2 and H_∞ approaches; this way, a robust H_2/H_∞ controller is obtained. Within the robust H_2/H_∞ controller, the weights of the H_2 and H_∞ control techniques are adjusted such that the aircraft accurately tracks the desired trajectory during the two main stages of the landing process (glide slope and flare). The new automatic landing system also consists of: a subsystem which models the geometry of landing, providing the imposed value of aircraft longitudinal velocity and altitude, an optimal observer for the estimation of aircraft state and a dynamic compensator providing one of the two components of the mixed H_2/H_∞ control law. The theoretical results are validated by numerical simulations for the landing of a Boeing 747; the results are very promising and prove the robustness of the new auto-landing system even in the presence of disturbances.

- Lungu, M., Lungu, R. *Landing Autopilot for the Control of Airplane by using the H-inf Control*. 12th Int. Conference on Aerospace, Robotics, Mechatronics, Mechanical Engineering, Manufacturing systems, Neurorehabilitation and Bioengineering (OPTIROB) 29 Iunie-3 Iulie 2017, Jupiter. International Journal of Modeling and Optimization (IJMO), vol. 7, nr. 3, 2017. *Databases: Google Scholar, EI (INSPEC, IET), Engineering & Technology Digital Library, Crossref, DOAJ.*

The paper focuses on automatic control of airplane in longitudinal and lateral-directional planes, during landing, by using the linearized dynamics of airplane and the H-inf control; both planes (longitudinal and lateral-directional) are treated; the new obtained automatic landing system will consist of three subsystems: the first two control airplane's motion in longitudinal plane, while the third one is used for the control of airplane motion in lateral-directional plane. The theoretical results are validated through numerical simulations for a Boeing 747 airplane.

- Voicu, S., Buţu, F. *H-Infinity Design for Automatic Landing System*. 12th International Conference on Aerospace, Robotics, Mechatronics, Mechanical Engineering, Manufacturing systems, Neurorehabilitation and Bioengineering (OPTIROB) 29 Iunie-3 Iulie 2017, Jupiter. International Journal of Modeling and Optimization (IJMO), vol. 7, nr. 3, 2017. *Databases: Google Scholar, EI (INSPEC, IET), Engineering & Technology Digital Library, Crossref, DOAJ.*

This study focuses on the H-inf control of aircraft during landing, our aim being also to design a robust control law for a wind shear profile using the H-inf control. This method is based on minimizing a closed loop transfer function norm in order to obtain the stability and robust performance of the system. Some of the design objectives are suggested in other works, but the numerical case study and the numerical implementation of the design procedure are performed and presented in this paper for the first time.

Also it worth mentioning that some of the information obtained from this phase of the project were used to improve the Master course ***„Aircraft automatic flight control during landing”***, University of Craiova. **Taking into account all the issues presed above, for Phase III of the project (*Design, validation and optimizing of the adaptive Auto Landing System; comparative studies; comparative studies between the two designed complete ALSs*), we consider that all the nine specific activities were fully acomplished and all the 11 specific objectives of Phase III were achieved.**

References (phase III of the project)

[1] Lungu, R., Lungu, M. *Application of H_2/H_∞ and Dynamic Inversion Techniques to Aircraft Landing Control*. Aerospace Science and Technology, vol. 46, pp. 146-158, 2015.

[2] Lungu, R., Lungu, M. *Automatic Control of Aircraft in Lateral-Directional Plane During Landing*. Asian Journal of Control, vol. 18, no. 3, pp. 433-446, 2016.

[3] Lungu, R., Lungu, M., Grigorie, T.L. *ALSs with conventional and fuzzy controllers considering wind shears and gyro errors*. Journal of Aerospace Engineering, 2012; 26 (4): 794-813.

[4] Salih, A., Zhahir A, Ariff, O.K., Amham, M. *Modeling and Simulation of a High Accurate Aircraft Ground-based Positioning and Landing System*. Proceeding of the IEEE International Conference on Space Science and Communication (IconSpace), 1-3 July 2013, Melaka, Malaysia.

[5] Lungu, M., Lungu, R. *Landing Auto-pilots for Airplane Motion in Longitudinal Plane using Adaptive Control Laws Based on Neural Networks and Dynamic Inversion*, Asian Journal of Control, vol. 19, no. 1, pp. 1-15, 2017.

[6] Calise, A., Hovakymyan, N., Idan, M. *Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks*, Automatica, vol. 37, no. 8, pp. 1201-1211, 2001.

[7] Lungu, M. *Flight control systems (Sisteme de conducere a zborului)*, Sitech Publisher, Craiova, 2008.

[8] Xu, K., Zhang, G., Xu, Y. *Intelligent Landing Control System for Civil Aviation Airplane with Dual Fuzzy Neural Network*, 8th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 171-175, 2011.

[9] Calise, A., Lee, S., Sharma, M. *Direct Adaptive Reconfigurable Control of a Tailless Fighter Airplane*, Rev. American Institute of Aeronautics and Astronautics, Georgia, USA, 2000.

[10] Chwa, D., Choi, J. *Adaptive Nonlinear Guidance Law Considering Control Loop Dynamics*, IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 4, pp. 1134-1143, 2003.

[11] Lungu, M., Lungu, R. *Automatic Control of Airplane Lateral-directional Motion during Landing using Neural Networks and Radio-technical Subsystems*. Neurocomputing Journal, vol. 171, pp. 471-481, 2016.

[12] Lungu, M., Lungu, R., Grigorie, L. *Automatic Command Systems for the Flight Direction Control during the Landing Process*. Proc. of Int. Symposium on Logistics and Industrial Informatics, August 25-27, 2011, Budapest, pp. 117-122.

[13] Lungu, R., Lungu, M., and Grigorie, T. L. *Automatic Control of Aircraft in Longitudinal Plane during Landing*. IEEE Transaction on Aerospace and Electronic Systems, vol. 49, no. 2, 2013, pp. 1338-1350.

APPENDIX

Appendix 1

```
close all; clear all;
% Faza 1 a controlului H2/H_inf la aterizare a avionului Boeing-747
for kk=1:2
   if kk==1
      e=zeros(7,1);
   else
       e=1*[0.1 0.1 0 0.1 0 0.1 0.1]';
   end
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';       % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;
a32=-0.164;a33=-0.412;a34=0;b11=0.1;b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
V0=70;Vxb=V0;Hp=420;Hp_deriv=0;gama_c=-2.5;alfa0=-0.5;teta0=-2.5;wy0=2;delta_p0=-3;delta_t0=2;Vx0=71;
Xp0=-Hp/tan(gama_c*pi/180);x0=[Vx0/V0;alfa0;wy0;teta0;Hp/V0;Hp_deriv/V0;delta_p0;delta_t0];xc0=x0;dxc0=x0-xc0;
wy_g=0;Vx_g=-67.6;alfa_g=0.05;teta_g=2.5;w=[Vx_g/V0;alfa_g;wy_g;teta_g];
Tt=.9;Tp=.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;k=0.5;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];
G=[a11 a21 V0*a31 0 0 -a21 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 1-a23/V0 0 0;a14/V0 a24/V0 a34 0 1 -a24/V0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[c2 0];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
D22=1*eye(7,7);Cr=[0 0 0 0 1 0 0 0];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);    %Q0=C0'*C0;
ham1=[A -B*pinv(R0)*B';-Q0 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);              %Q1=C1'*C1;R1=D11'*D11;
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);
P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
P_c=(1-k)*P+k*P_inf;Ps_c=(1-k)*Ps+k*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
% Caracteristici grafice
sim('Panta_ALS6_sch');
if kk==2
   Vx_er=Vx;Teta_er=Teta;wy_er=wy;gama_er=gama;Alfa_er=Alfa;Delta_p_er=Delta_p;Delta_t_er=Delta_t;
   H_ref_er=H_ref;teta_b_er=teta_b;Vz_er=Vz;H_er=H;Vx_b_er=Vx_b;H1_er=H1;
end
disp('------------------------------------------------------------------------------------------------------------');
end
subplot(4,3,1);plot(t,Vx(1,:),t,Vx_er(1,:),'r--');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(4,3,2);plot(t,Vz(1,:)/2,t,Vz_er(1,:)/2,'r--');grid;ylabel('Vz [m/s]');xlabel('Timp [s]');
subplot(4,3,3);plot(t,wy(1,:),t,wy_er(1,:),'r--');grid;ylabel('Omegay [grd/s]');xlabel('Timp [s]');
subplot(4,3,4);plot(t,Teta(1,:),t,Teta_er(1,:),'r--');grid;ylabel('Teta [grd]');xlabel('Timp [s]');
subplot(4,3,5);plot(t,Alfa(1,:),t,Alfa_er(1,:),'r--');grid;ylabel('Alfa [grd]');xlabel('Timp [s]');
subplot(4,3,6);plot(t,gama(1,:),t,gama_er(1,:),'r--');grid;ylabel('gama [grd]');xlabel('Timp [s]');
subplot(4,3,7);plot(t,H(1,:),t,H_er(1,:),'r--');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(4,3,8);plot(t,Delta_p(1,:),t,Delta_p_er(1,:),'r--');grid;ylabel('Bracaj profundor [grd]');xlabel('Timp [s]');
subplot(4,3,9);plot(t,Delta_t(1,:),t,Delta_t_er(1,:),'r--');grid;ylabel('Comanda maneta gaze [grd]');xlabel('Timp [s]');
subplot(4,3,10);plot(t,-Vx(1,:)+V0*ones(1,length(Vx)),t,-Vx_er(1,:)+V0*ones(1,length(Vx)),'r--');grid;
ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(4,3,11);plot(t,-gama(1,:)+gama_c*ones(1,length(gama)),t,-gama_er(1,:)+gama_c*ones(1,length(gama)),'r--');grid;
```

```
ylabel('gama_c-gama [grd');xlabel('Timp [s]');
subplot(4,3,12);plot(t,(-H1(1,:)+H_ref(1,:)),t,(-H1_er(1,:)+H_ref_er(1,:)),'r--');grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');

% -----------------------------------------------------------------------------------------------------------------------------------------------------------

clear all;
% Faza 2 a controlului H2/H_inf la aterizare a avionului Boeing-747
for kk=1:2
   if kk==1
      e=zeros(7,1);
   else
      e=1*[0.1 0.1 0 0.1 0 0.1 0.1]';
   end
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';       % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;
a32=-0.164;a33=-0.412;a34=0;b11=0.1;b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
gama_c=0;V0=70;Vxb=V0;Hp_int=0;H0=30;Tau=4;alfa0=0;teta0=0.1;wy0=0;delta_p0=-2;delta_t0=0;Vx0=V0;
x0=[Vx0/V0;alfa0;wy0;teta0;H0/V0;Hp_int/V0;delta_p0;delta_t0];xc0=0;dxc0=x0-xc0;
wy_g=0;Vx_g=-79.75;alfa_g=0;w=[Vx_g/V0;alfa_g;wy_g];Tt=0.9;Tp=0.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;k=0.5;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];
G=[a11 a21 V0*a31 0 0 0 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[0 c2];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
D22=1*eye(7,7);Cr=[0 0 0 0 1 0 0];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);ham1=[A -B*pinv(R0)*B';-Q0 -A'];
[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);                      % !!! Q1 si R1
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);
P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
P_c=(1-k)*P+k*P_inf;Ps_c=(1-k)*Ps+k*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
  V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
% Caracteristici grafice
sim('Planare_ALS6_sch');
if kk==2
   Vx_er=Vx;Teta_er=Teta;wy_er=wy;gama_er=gama;Alfa_er=Alfa;Delta_p_er=Delta_p;Delta_t_er=Delta_t;
   H_ref_er=H_ref;teta_b_er=teta_b;Vz_er=Vz;H_er=H;Vx_b_er=Vx_b;
end
disp('------------------------------------------------------------------------------------------------------------');
end
h=figure;
subplot(4,3,1);plot(t,Vx(1,:),t,Vx_er(1,:),'r--');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(4,3,2);plot(t,-Vz(1,:)/4,t,-Vz_er(1,:)/4,'r--');grid;ylabel('Vz [m/s]');xlabel('Timp [s]');
subplot(4,3,3);plot(t,wy(1,:),t,wy_er(1,:),'r--');grid;ylabel('Omegay [grd/s]');xlabel('Timp [s]');
subplot(4,3,4);plot(t,Teta(1,:),t,Teta_er(1,:),'r--');grid;ylabel('Teta [grd]');xlabel('Timp [s]');
subplot(4,3,5);plot(t,Alfa(1,:),t,Alfa_er(1,:),'r--');grid;ylabel('Alfa [grd]');xlabel('Timp [s]');
subplot(4,3,6);plot(t,gama(1,:),t,gama_er(1,:),'r--');grid;ylabel('gama [grd]');xlabel('Timp [s]');
subplot(4,3,7);plot(t,H(1,:),t,H_er(1,:),'r--');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(4,3,8);plot(t,Delta_p(1,:),t,Delta_p_er(1,:),'r--');grid;ylabel('Bracaj profundor [grd]');xlabel('Timp [s]');
subplot(4,3,9);plot(t,Delta_t(1,:),t,Delta_t_er(1,:),'r--');grid;ylabel('Comanda maneta gaze [grd]');xlabel('Timp [s]');
subplot(4,3,10);plot(t,-Vx(1,:)+V0*ones(1,length(Vx)),t,-Vx_er(1,:)+V0*ones(1,length(Vx)),'r--');grid;
ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(4,3,11);plot(t,-gama(1,:)+gama_c*ones(1,length(gama)),t,-gama_er(1,:)+gama_c*ones(1,length(gama)),'r--');grid;
ylabel('gama_c-gama [grd');xlabel('Timp [s]');
subplot(4,3,12);plot(t,(H_ref'-H(1,:)),t,(H_ref_er'-H_er(1,:)),'r--');grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
```

```
% ---------------------------------------------------------------------------------------------------------------------------------------------------
% Controlul aterizarii - miscarea laterala
clear all;
% Modelul matematic al avionului Charlie (miscarea laterala)
for k=1:2
   if k==2
      e=zeros(7,1);
   else
      e=10*[0 0 0.1 0 0.1 0 0.1]';
   end
for jj=5:-1:1
   if jj==1
     Vvy0=2;
   elseif jj==2
     Vvy0=4;
   elseif jj==3
     Vvy0=6;
   elseif jj==4
     Vvy0=8;
   else
     Vvy0=10;
   end
Y_c=0;Yb0=1;Beta_b_0=0.1;Beta_c=0;Te=0.7;Td=0.1;V0=67;w1=1;w2=1;p=25;csi1=0.7;csi2=0.7;L21=1;b01=1;b02=1;L12=0;
beta0=0.1;wx0=0;wz0=-2;fi0=0;psi0=0.1;y0=25;delta_e0=0;delta_d0=0;x0=[beta0;wx0;wz0;fi0;psi0;y0;delta_e0;delta_d0];
miu1=1;miu2=1;c1=sqrt(0.01);c2=sqrt(0.01);xc0=x0;dx_c0=xc0;const=2;T=20;M1=-5;M2=5;
C0=[0 0 0 0 0 1 0 0];C1=[1 0 0 0 0 0 0 0];D01=[c1 0];D11=[0 c2];
a11=-0.089/V0;a12=0;a13=-1;a14=0.15;b11=0.001;b12=0.015;a21=-1.33;a22=-0.98;a23=0.33;
b21=0.23;b22=0.06;a31=0.17;a32=-0.17;a33=-0.217;b31=0.026;b32=-0.15;
A=[a11 a12 a13 a14 0 0 b11 b12;a21 a22 a23 0 0 0 b21 b22;a31 a32 a33 0 0 0 b31 b32;
   0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;-V0 0 0 0 V0 0 0 0;0 0 0 0 0 0 -1/Te 0;0 0 0 0 0 0 0 -1/Td];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Te 0;0 1/Td];G=[a11/V0;a21/V0;a31/V0;0;0;1;0;0];
C=[0 0 0 0 0 1 0 0;-V0 0 0 0 V0 0 0 0;1 0 0 0 0 0 0 0;0 0 0 1 0 0 0 0;0 1 0 0 0 0 0 0;0 0 0 0 1 0 0 0;0 0 1 0 0 0 0 0];
Cp=[0 0 0 0 0 1 0 0;1 0 0 0 0 0 0 0];
% Calculul matricelor algoritmului
a11p=a11*a11+a12*a21+a13*a31;a12p=a11*a12+a12*a22+a13*a32+a14;a13p=a11*a13+a12*a23+a13*a33;a14p=a11*a14;
a17p=a11*b11+a12*b21+a13*b31-b11/Te;a18p=a11*b12+a12*b22+a13*b32-b12/Td;a61p=V0*(a31-a11p);a62p=V0*(a32-a12p);
a63p=V0*(a33-a13p);a64p=-V0*a14p;a67p=V0*(b31-a17p);a68p=V0*(b32-a18p);a31p=a31-a11p;a61s=a61p-V0*a11;
a62s=a62p-V0*a12;a63s=a63p-V0*(a13+1);a64s=a64p-V0*a14;a67s=a67p-V0*b11;a68s=a68p-V0*b12;a31s=a31p+1;
A_x=[a61p a62p a63p a64p 0 0 a67p a68p;a11p a12p a13p a14p 0 0 a17p a18p];
AA=[a62s a63s a64s 0 a67s a68s;a12p a13p a14p 0 a17p a18p];B_u=[-V0*b11/Te -V0*b12/Td;b11/Te b12/Td];
G_p=[a31p -a11;-a11p/V0 a11/V0];G_s=[a31s -a11;-a11p/V0 a11/V0];
D22=200*eye(7,7);D_b=[D01;D11];R1=D_b'*D_b;C_b=[C0;C1];Q1=C_b'*C_b;
% Rezolvare ecuatie Ricatti 1
ham1=[A -B*inv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);
P_inf=x2*inv(x1);K_inf=(inv(R1))*B'*P_inf;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C+(miu2^(-2))*C_b'*C_b;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);
Ps_inf=x2*inv(x1);L_inf=Ps_inf*C'*(inv(D22'*D22));
for i=1:size(K_inf,1)
   for j=1:size(K_inf,2)
      KK_inf(i,j)=real(K_inf(i,j));
   end
end
for i=1:size(L_inf,1)
   for j=1:size(L_inf,2)
      LL_inf(i,j)=real(L_inf(i,j));
   end
end
K_inf=KK_inf;L_inf=LL_inf;
% Rezolvare sistemului cu 3 ec si 2 nec. (kd1 si kp1) - ec. caracterisitica 1
s1=-0.1;s2=-0.2;s3=-0.3;   % solutiile impuse ec. caracteristice 1
A1=b01*[s1 1;s2 1;s3 1];B1=[-(s1^3)-L21*(s1^2);-(s2^3)-L21*(s2^2);-(s3^3)-L21*(s3^2)];
x1=lsqr(A1,B1);kd1=x1(1,:);kp1=x1(2,:);rad_1=roots([1 L21 b01*kd1 b01*kp1])
% Rezolvare sistemului cu 2 ec si 2 nec. (kd2 si kp2) - ec. caracterisitica 2
s1=-1.1;s2=-0.7;     % solutiile impuse ec. caracteristice 2
A2=b02*[s1 1;s2 1];B2=[-s1^2-L12*s1;-s2^2-L12*s2];
x2=lsqr(A2,B2);kd2=x2(1,:);kp2=x2(2,:);rad_2=roots([1 (b02*kd2+L12) (b02*kp2)])
% Caracteristici grafice
sim('S2_sch');
YY(jj,:)=Y(1,:);BB(jj,:)=betta(1,:);
end    % end for jj
```

```
if k==1
   Err_er=Err;betta_er=betta;wx_er=wx;wz_er=wz;fi_er=fi;csi_er=csi;Y_er=Y;delta_e_er=delta_e;delta_d_er=delta_d;
   Y_b_er=Y_b;Beta_b_er=Beta_b;u1_er=u1;u2_er=u2;ub1_er=ub1;ub2_er=ub2;u_inf_1_er=u_inf_1;u_inf_2_er=u_inf_2;
end
disp('---------------------------------------------------------------------------------------------------------');
end     % end for k
h=figure;
plot(t,Err(1,:),'b');hold on;plot(t,Err(2,:),'g');hold on;plot(t,Err(3,:),'r');hold on;plot(t,Err(4,:),'c');hold on;
plot(t,Err(5,:),'m');hold on;plot(t,Err(6,:),'y');hold on;plot(t,Err(7,:),'k');plot(t,Err(8,:),'b--');grid;
ylabel('Erorile observerului fara erori ale senzorilor');xlabel('Timp [s]');
h=figure;
subplot(4,2,1);plot(t,betta(1,:),t,betta_er(1,:),'r--');grid;ylabel('Beta [grd]');xlabel('Timp [s]');
subplot(4,2,2);plot(t,wx(1,:),t,wx_er(1,:),'r--');grid;ylabel('Omegax [grd/s]');xlabel('Timp [s]');
subplot(4,2,3);plot(t,wz(1,:),t,wz_er(1,:),'r--');grid;ylabel('Omegaz [grd/s]');xlabel('Timp [s]');
subplot(4,2,4);plot(t,fi(1,:),t,fi_er(1,:),'r--');grid;ylabel('Fi [grd]');xlabel('Timp [s]');
subplot(4,2,5);plot(t,csi(1,:),t,csi_er(1,:),'r--');grid;ylabel('Psi [grd]');xlabel('Timp [s]');
subplot(4,2,6);plot(t,Y(1,:),t,Y_er(1,:),'r--');grid;ylabel('Abaterea laterala [m]');xlabel('Timp [s]');
subplot(4,2,7);plot(t,delta_e(1,:),t,delta_e_er(1,:),'r--');grid;ylabel('Bracaj eleroane [grd]');xlabel('Timp [s]');
subplot(4,2,8);plot(t,delta_d(1,:),t,delta_d_er(1,:),'r--');grid;ylabel('Bracaj directie [grd]');xlabel('Timp [s]');
h=figure;plot(t,YY(1,:),'b',t,YY(2,:),'g',t,YY(3,:),'r',t,YY(4,:),'m',t,YY(5,:),'k');grid;ylabel('Abaterea laterala [m]');xlabel('Timp [s]');
h=figure;plot(t,BB(1,:),'b',t,BB(2,:),'g',t,BB(3,:),'r',t,BB(4,:),'m',t,BB(5,:),'k');grid;ylabel('Beta [grd]');xlabel('Timp [s]');
```

Appendix 2

```
close all; clear all;
% Faza 1 a controlului H2/H_inf la aterizare a avionului Boeing-747
f=[1 1.1 1.3 1.4 1.6];e=[0.1 0.1 0 0.1 0 0.1 0.1]';
for jj=1:5
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';       % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;
a32=-0.164;a33=-0.412;a34=0;b11=0.1;b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
V0=70;Vxb=V0;Hp=420;Hp_deriv=0;gama_c=-2.5;alfa0=-0.5;teta0=-2.5;wy0=2;delta_p0=-3;delta_t0=2;Vx0=71;
Xp0=-Hp/tan(gama_c*pi/180);x0=[Vx0/V0;alfa0;wy0;teta0;Hp/V0;Hp_deriv/V0;delta_p0;delta_t0];xc0=x0;dxc0=x0-xc0;
wy_g=0;Vx_g=-67.6;alfa_g=0.05;teta_g=2.5;w=[Vx_g/V0;alfa_g;wy_g;teta_g];
Tt=.9;Tp=.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;k=0.5;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];
G=[a11 a21 V0*a31 0 0 -a21 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 1-a23/V0 0 0;a14/V0 a24/V0 a34 0 1 -a24/V0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[c2 0];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
D22=f(jj)*eye(7,7);Cr=[0 0 0 0 1 0 0 0];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);    %Q0=C0'*C0;
ham1=[A -B*pinv(R0)*B';-Q0 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);              %Q1=C1'*C1;R1=D11'*D11;
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);
P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
P_c=(1-k)*P+k*P_inf;Ps_c=(1-k)*Ps+k*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
   V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
% Caracteristici grafice
sim('Panta_ALS6_sch');
H_D22(jj,:)=H(1,:);ALT(jj,:)=H1(1,:);Vx_D22(jj,:)=Vx(1,:);gama_D22(jj,:)=gama(1,:);teta_D22(jj,:)=Teta(1,:);tetac_D22(jj,:)=tetac(1,:);
disp('---------------------------------------------------------------------------------------------------------');
end   % end jj
subplot(2,2,1);plot(t,H_D22(1,:),'b',t,H_D22(2,:),'r',t,H_D22(3,:),'k',t,H_D22(4,:),'c',t,H_D22(5,:),'m');grid;ylabel('H[m]');xlabel('Timp [s]');
```

```
subplot(2,2,2);plot(t,(H_ref(1,:)-ALT(1,:)),'b',t,(H_ref(1,:)-ALT(2,:)),'r',t,(H_ref(1,:)-ALT(3,:)),'k',t,(H_ref(1,:)-ALT(4,:)),'c',t,(H_ref(1,:)-
ALT(5,:)),'m');grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
subplot(2,2,3);plot(t,V0*ones(1,length(Vx))-Vx_D22(1,:),'b',t,V0*ones(1,length(Vx))-Vx_D22(2,:),'r',t,V0*ones(1,length(Vx))-
Vx_D22(3,:),'k',t',V0*ones(1,length(Vx))-Vx_D22(4,:),'c',t,V0*ones(1,length(Vx))-Vx_D22(5,:),'m');
grid;ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(2,2,4);plot(t,gama_c*ones(1,length(gama))-gama_D22(1,:),'b',t,gama_c*ones(1,length(gama))-
gama_D22(2,:),'r',t,gama_c*ones(1,length(gama))-gama_D22(3,:),'k',t',gama_c*ones(1,length(gama))-
gama_D22(4,:),'c',t,gama_c*ones(1,length(gama))-gama_D22(5,:),'m');
grid;ylabel('gama_c-gama [grd');xlabel('Timp [s]');
h=figure;
subplot(1,2,1);plot(t,Vx_D22(1,:),'b',t,Vx_D22(2,:),'r',t,Vx_D22(3,:),'k',t',Vx_D22(4,:),'c',t,Vx_D22(5,:),'m');
grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(1,2,2);plot(t,gama_D22(1,:),'b',t,gama_D22(2,:),'r',t,gama_D22(3,:),'k',t',gama_D22(4,:),'c',t,gama_D22(5,:),'m');
grid;ylabel('gama [grd');xlabel('Timp [s]');
% ------------------------------------------------------------------------------------------------------------------------------------------------
clear all;
% Faza 2 a controlului H2/H_inf la aterizare a avionului Boeing-747
f=[1 1.1 1.3 1.4 1.6];e=[0.1 0.1 0 0.1 0 0.1 0.1]';
for jj=1:5
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';      % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;a32=-0.164;a33=-0.412;a34=0;b11=0.1;
b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
gama_c=0;V0=70;Vxb=V0;Hp_int=0;H0=30;Tau=4;alfa0=0;teta0=0.1;wy0=0;delta_p0=-2;delta_t0=0;Vx0=V0;
x0=[Vx0/V0;alfa0;wy0;teta0;H0/V0;Hp_int/V0;delta_p0;delta_t0];xc0=0;dxc0=x0-xc0;
wy_g=0;Vx_g=-79.75;alfa_g=0;w=[Vx_g/V0;alfa_g;wy_g];Tt=0.9;Tp=0.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;k=0.5;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];
G=[a11 a21 V0*a31 0 0 0 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[0 c2];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
D22=f(jj)*eye(7,7);Cr=[0 0 0 0 1 0 0];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);ham1=[A -B*pinv(R0)*B';-Q0 -A'];
[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);                    % !!! Q1 si R1
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);
P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
P_c=(1-k)*P+k*P_inf;Ps_c=(1-k)*Ps+k*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
   V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
% Caracteristici grafice
sim('Planare_ALS6_sch');
H_D22(jj,:)=H(1,:);ALT(jj,:)=H(1,:);Vx_D22(jj,:)=Vx(1,:);gama_D22(jj,:)=gama(1,:);teta_D22(jj,:)=Teta(1,:);tetac_D22(jj,:)=tetac(1,:);
disp('-----------------------------------------------------------------------------------------------------------');
end   % end jj
h=figure;
subplot(2,2,1);plot(t,H_D22(1,:),'b',t,H_D22(2,:),'r',t,H_D22(3,:),'k',t,H_D22(4,:),'c',t,H_D22(5,:),'m');
grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(2,2,2);plot(t,(H_ref'-ALT(1,:)),'b',t,(H_ref'-ALT(2,:)),'r',t,(H_ref'-ALT(3,:)),'k',t,(H_ref'-ALT(4,:)),'c',t,(H_ref'-ALT(5,:)),'m');
grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
subplot(2,2,3);plot(t,V0*ones(1,length(Vx))-Vx_D22(1,:),'b',t,V0*ones(1,length(Vx))-Vx_D22(2,:),'r',t,V0*ones(1,length(Vx))-
Vx_D22(3,:),'k',t',V0*ones(1,length(Vx))-Vx_D22(4,:),'c',t,V0*ones(1,length(Vx))-Vx_D22(5,:),'m');
grid;ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(2,2,4);plot(t,gama_c*ones(1,length(gama))-gama_D22(1,:),'b',t,gama_c*ones(1,length(gama))-
gama_D22(2,:),'r',t,gama_c*ones(1,length(gama))-gama_D22(3,:),'k',t',gama_c*ones(1,length(gama))-
gama_D22(4,:),'c',t,gama_c*ones(1,length(gama))-gama_D22(5,:),'m');
```

```
grid;ylabel('gama_c-gama [grd');xlabel('Timp [s]'); h=figure; subplot(1,2,1);
plot(t,Vx_D22(1,:),'b',t,Vx_D22(2,:),'r',t,Vx_D22(3,:),'k',t',Vx_D22(4,:),'c',t,Vx_D22(5,:),'m');grid; ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(1,2,2);plot(t,gama_D22(1,:),'b',t,gama_D22(2,:),'r',t,gama_D22(3,:),'k',t',gama_D22(4,:),'c',t,gama_D22(5,:),'m');
grid;ylabel('gama [grd');xlabel('Timp [s]');
```

Appendix 3

```
% Comparatie intre sistemul fara PCH si acelasi sistem cu PCH (miscarea laterala)
close all;clear all;
% Modelul matematic al avionului Charlie (miscarea laterala)
for const=1:2     % const=2 pentru vh=0 si const=1 pentru vh~=0
for k=1:1     % simularea se face doar in prezenta erorilor senzorilor
   if k==2
      e=zeros(7,1);
   else
      e=10*[0 0 0.1 0 0.1 0 0.1]';
   end
for jj=1:1:5
   if jj==1
     Vvy0=2;
   elseif jj==2
     Vvy0=4;
   elseif jj==3
     Vvy0=6;
   elseif jj==4
     Vvy0=8;
   else
     Vvy0=10;
   end
Y_c=0;Yb0=1;Beta_b_0=0.1;Beta_c=0;Te=0.7;Td=0.1;V0=67;w1=1;w2=1;p=25;csi1=0.7;csi2=0.7;L21=1;b01=1;b02=1;L12=0;
beta0=0.1;wx0=0;wz0=-2;fi0=0;psi0=0.1;y0=25;delta_e0=0;delta_d0=0;x0=[beta0;wx0;wz0;fi0;psi0;y0;delta_e0;delta_d0];
miu1=1;miu2=1;c1=sqrt(0.01);c2=sqrt(0.01);xc0=x0;dx_c0=xc0;T=20;M1=-5;M2=5; C0=[0 0 0 0 0 1 0 0];C1=[1 0 0 0 0 0 0 0];
D01=[c1 0];D11=[0 c2]; a11=-0.089/V0;a12=0;a13=-1;a14=0.15;b11=0.001;b12=0.015;a21=-1.33;a22=-0.98;a23=0.33;
b21=0.23;b22=0.06;a31=0.17;a32=-0.17;a33=-0.217;b31=0.026;b32=-0.15;
A=[a11 a12 a13 a14 0 0 b11 b12;a21 a22 a23 0 0 0 b21 b22;a31 a32 a33 0 0 0 b31 b32;
   0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;-V0 0 0 0 V0 0 0 0;0 0 0 0 0 0 -1/Te 0;0 0 0 0 0 0 0 -1/Td];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Te 0;0 1/Td];G=[a11/V0;a21/V0;a31/V0;0;0;1;0;0]; Cp=[0 0 0 0 0 1 0 0;1 0 0 0 0 0 0 0];
C=[0 0 0 0 0 1 0 0;-V0 0 0 0 V0 0 0 0;1 0 0 0 0 0 0 0;0 0 0 1 0 0 0 0;0 1 0 0 0 0 0 0;0 0 0 0 1 0 0 0;0 0 1 0 0 0 0 0];
% Calculul matricelor algoritmului
a11p=a11*a11+a12*a21+a13*a31;a12p=a11*a12+a12*a22+a13*a32+a14;a13p=a11*a13+a12*a23+a13*a33;a14p=a11*a14;
a17p=a11*b11+a12*b21+a13*b31-b11/Te;a18p=a11*b12+a12*b22+a13*b32-b12/Td;a61p=V0*(a31-a11p);
a62p=V0*(a32-a12p);a63p=V0*(a33-a13p);a64p=-V0*a14p;a67p=V0*(b31-a17p);a68p=V0*(b32-a18p);a31p=a31-a11p;
a61s=a61p-V0*a11;a62s=a62p-V0*a12;a63s=a63p-V0*(a13+1);a64s=a64p-V0*a14;a67s=a67p-V0*b11;
a68s=a68p-V0*b12;a31s=a31p+1;A_x=[a61p a62p a63p a64p 0 0 a67p a68p;a11p a12p a13p a14p 0 0 a17p a18p];
AA=[a62s a63s a64s 0 a67s a68s;a12p a13p a14p 0 a17p a18p];B_u=[-V0*b11/Te -V0*b12/Td;b11/Te b12/Td];
G_p=[a31p -a11;-a11p/V0 a11/V0];G_s=[a31s -a11;-a11p/V0 a11/V0];
D22=200*eye(7,7);D_b=[D01;D11];R1=D_b'*D_b;C_b=[C0;C1];Q1=C_b'*C_b;
% Rezolvare ecuatie Ricatti 1
ham1=[A -B*inv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);
P_inf=x2*inv(x1);K_inf=(inv(R1))*B'*P_inf;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C+(miu2^(-2))*C_b'*C_b;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);
Ps_inf=x2*inv(x1);L_inf=Ps_inf*C'*(inv(D22'*D22));
for i=1:size(K_inf,1)
   for j=1:size(K_inf,2)
      KK_inf(i,j)=real(K_inf(i,j));
   end
end
for i=1:size(L_inf,1)
   for j=1:size(L_inf,2)
      LL_inf(i,j)=real(L_inf(i,j));
   end
end
K_inf=KK_inf;L_inf=LL_inf;
eig(A-B*K_inf)
eig(A-L_inf*C)
% Rezolvare sistemului cu 3 ec si 2 nec. (kd1 si kp1) - ec. caracterisitica 1
s1=-0.1;s2=-0.2;s3=-0.3;   % solutiile impuse ec. caracteristice 1
A1=b01*[s1 1;s2 1;s3 1];B1=[-(s1^3)-L21*(s1^2);-(s2^3)-L21*(s2^2);-(s3^3)-L21*(s3^2)];
x1=lsqr(A1,B1);kd1=x1(1,:);kp1=x1(2,:);rad_1=roots([1 L21 b01*kd1 b01*kp1])
```

```
% Rezolvare sistemului cu 2 ec si 2 nec. (kd2 si kp2) - ec. caracterisitica 2
s1=-1.1;s2=-0.7;     % solutiile impuse ec. caracteristice 2
A2=b02*[s1 1;s2 1];B2=[-s1^2-L12*s1;-s2^2-L12*s2];x2=lsqr(A2,B2);kd2=x2(1,:);kp2=x2(2,:);rad_2=roots([1 (b02*kd2+L12) (b02*kp2)])
% Caracteristici grafice
sim('S2_sch');
if const==2
   YY(jj,:)=Y(1,:);BB(jj,:)=betta(1,:);
else
   YY_vh(jj,:)=Y(1,:);BB_vh(jj,:)=betta(1,:);
   betta_vh=betta;wx_vh=wx;wz_vh=wz;fi_vh=fi;csi_vh=csi;Y_vh=Y;delta_e_vh=delta_e;delta_d_vh=delta_d;
   Y_b_vh=Y_b;Beta_b_vh=Beta_b;u1_vh=u1;u2_vh=u2;ub1_vh=ub1;ub2_vh=ub2;u_inf_1_vh=u_inf_1;u_inf_2_vh=u_inf_2;
end   % end if const
end   % end for jj
if k==1
    Err_er=Err;betta_er=betta;wx_er=wx;wz_er=wz;fi_er=fi;csi_er=csi;Y_er=Y;delta_e_er=delta_e;delta_d_er=delta_d;
    Y_b_er=Y_b;Beta_b_er=Beta_b;u1_er=u1;u2_er=u2;ub1_er=ub1;ub2_er=ub2;u_inf_1_er=u_inf_1;u_inf_2_er=u_inf_2;
end
disp('----------------------------------------------------------------------------------------------------------------');
end      % end for k
end      % for const
h=figure;    % Caracteristici cu si fara PCH pentru Vvy=10 m/s
subplot(4,2,1);plot(t,betta(1,:),t,betta_vh(1,:),'r--');grid;ylabel('Beta [grd]');xlabel('Timp [s]');
subplot(4,2,2);plot(t,wx(1,:),t,wx_vh(1,:),'r--');grid;ylabel('Omegax [grd/s]');xlabel('Timp [s]');
subplot(4,2,3);plot(t,wz(1,:),t,wz_vh(1,:),'r--');grid;ylabel('Omegaz [grd/s]');xlabel('Timp [s]');
subplot(4,2,4);plot(t,fi(1,:),t,fi_vh(1,:),'r--');grid;ylabel('Fi [grd]');xlabel('Timp [s]');
subplot(4,2,5);plot(t,csi(1,:),t,csi_vh(1,:),'r--');grid;ylabel('Psi [grd]');xlabel('Timp [s]');
subplot(4,2,6);plot(t,Y(1,:),t,Y_vh(1,:),'r--');grid;ylabel('Abaterea laterala [m]');xlabel('Timp [s]');
subplot(4,2,7);plot(t,delta_e(1,:),t,delta_e_vh(1,:),'r--');grid;ylabel('Bracaj eleroane [grd]');xlabel('Timp [s]');
subplot(4,2,8);plot(t,delta_d(1,:),t,delta_d_vh(1,:),'r--');grid;ylabel('Bracaj directie [grd]');xlabel('Timp [s]');
h=figure;   % Caracteristici cu si fara PCH pentru Vvy=4 m/s, Vvy=6 m/s, Vvy=8 m/s, Vvy=10 m/s
subplot(2,2,1);plot(t,YY_vh(2,:),'r--',t,YY(2,:),'b');grid;ylabel('Y pt Vvy=4 m/s [m]');xlabel('Timp [s]');
subplot(2,2,2);plot(t,YY_vh(3,:),'r--',t,YY(3,:),'b');grid;ylabel('Y pt Vvy=6 m/s [m]');xlabel('Timp [s]');
subplot(2,2,3);plot(t,YY_vh(4,:),'r--',t,YY(4,:),'b');grid;ylabel('Y pt Vvy=8 m/s [m]');xlabel('Timp [s]');
subplot(2,2,4);plot(t,YY_vh(5,:),'r--',t,YY(5,:),'b');grid;ylabel('Y pt Vvy=10 m/s [m]');xlabel('Timp [s]');
```

Appendix 4

```
close all; clear all;
% Faza 1 a controlului H2/H_inf la aterizare a avionului Boeing-747
% Dependenta de constanta k
k=[0.1 0.5 0.9];e=[0.1 0.1 0 0.1 0 0.1 0.1]';
for jj=1:3
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';        % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;
a32=-0.164;a33=-0.412;a34=0;b11=0.1;b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
V0=70;Vxb=V0;Hp=420;Hp_deriv=0;gama_c=-2.5;alfa0=-0.5;teta0=-2.5;wy0=2;delta_p0=-3;delta_t0=2;Vx0=71;
Xp0=-Hp/tan(gama_c*pi/180);x0=[Vx0/V0;alfa0;wy0;teta0;Hp/V0;Hp_deriv/V0;delta_p0;delta_t0];xc0=x0;dxc0=x0-xc0;
wy_g=0;Vx_g=-67.6;alfa_g=0.05;teta_g=2.5;w=[Vx_g/V0;alfa_g;wy_g;teta_g];Tt=.9;Tp=.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];
G=[a11 a21 V0*a31 0 0 -a21 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 1-a23/V0 0 0;a14/V0 a24/V0 a34 0 1 -a24/V0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[c2 0];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
D22=1*eye(7,7);Cr=[0 0 0 0 1 0 0 0];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);    %Q0=C0'*C0;
ham1=[A -B*pinv(R0)*B';-Q0 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);
P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);
Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);              %Q1=C1'*C1;R1=D11'*D11;
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);
P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
```

```
P_c=(1-k(jj))*P+k(jj)*P_inf;Ps_c=(1-k(jj))*Ps+k(jj)*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
  V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
% Caracteristici grafice
sim('Panta_ALS6_sch');
H_k(jj,:)=H(1,:);ALT(jj,:)=H1(1,:);Vx_k(jj,:)=Vx(1,:);gama_k(jj,:)=gama(1,:);teta_k(jj,:)=Teta(1,:);tetac_k(jj,:)=tetac(1,:);
disp('--------------------------------------------------------------------------------------------------------------');
end   % end jj
subplot(2,2,1);plot(t,H_k(1,:),'b',t,H_k(2,:),'k',t,H_k(3,:),'r');
grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(2,2,2);plot(t,(H_ref(1,:)-ALT(1,:))/1500,'b',t,(H_ref(1,:)-ALT(2,:))/1500,'k',t,(H_ref(1,:)-ALT(3,:))/1500,'r');
grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
subplot(2,2,3);plot(t,V0*ones(1,length(Vx))-Vx_k(1,:),'b',t,V0*ones(1,length(Vx))-Vx_k(2,:),'k',t,V0*ones(1,length(Vx))-Vx_k(3,:),'r');
grid;ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(2,2,4);plot(t,gama_c*ones(1,length(gama))-gama_k(1,:),'b',t,gama_c*ones(1,length(gama))-
gama_k(2,:),'k',t,gama_c*ones(1,length(gama))-gama_k(3,:),'r');
grid;ylabel('gama_c-gama [grd');xlabel('Timp [s]');
% ----------------------------------------------------------------------------------------------------------------------------------------------------------
clear all;
% Faza 2 a controlului H2/H_inf la aterizare a avionului Boeing-747
k=[0.1 0.5 0.9];e=[0.1 0.1 0 0.1 0 0.1 0.1]';
for jj=1:3
w1=2;w2=2;p=25;csi1=0.7;csi2=0.7;ita0=-1*[.1 .2 .3]';        % ita0=randn(3,1)
a11=-0.021;a12=0.122;a14=-0.322;a21=-0.209;a22=-0.53;a23=2.21;a24=0;a31=0.017;
a32=-0.164;a33=-0.412;a34=0;b11=0.1;b12=1;b21=-0.064;b22=-0.044;b31=-0.378;b32=0.544;
gama_c=0;V0=70;Vxb=V0;Hp_int=0;H0=30;Tau=4;alfa0=0;teta0=0.1;wy0=0;delta_p0=-2;delta_t0=0;Vx0=V0;
x0=[Vx0/V0;alfa0;wy0;teta0;H0/V0;Hp_int/V0;delta_p0;delta_t0];xc0=0;dxc0=x0-xc0;
wy_g=0;Vx_g=-79.75;alfa_g=0;w=[Vx_g/V0;alfa_g;wy_g];Tt=0.9;Tp=0.9;c1=sqrt(0.1);c2=sqrt(10);miu1=50;miu2=100;
A=[a11 a12 0 a14/V0 0 0 b11/V0 b12/V0;a21 a22 a23/V0 a24/V0 0 0 b21/V0 b22/V0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;
0 0 1 0 0 0 0 0;0 -1 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 0 -1/Tp 0;0 0 0 0 0 0 0 -1/Tt];
B=[0 0;0 0;0 0;0 0;0 0;0 0;1/Tp 0;0 1/Tt];G=[a11 a21 V0*a31 0 0 0 0 0;a12 a22 V0*a32 0 -1 -a22 0 0;0 a23/V0 a33 1 0 0 0 0]';
C0=[0 0 0 1 0 0 0 0];D01=[c1 0];C1=[V0 0 0 0 0 0 0 0];D11=[0 c2]; D22=1*eye(7,7);Cr=[0 0 0 0 1 0 0];
C=[V0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 V0 0 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
% Rezolvare ecuatie Ricatti 1
R0=1*eye(2,2);Q0=10*eye(8,8);ham1=[A -B*pinv(R0)*B';-Q0 -A'];
[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham1);P=x2*inv(x1);P=real(P);K=(pinv(R0))*B'*P;
% Rezolvare ecuatie Ricatti 2
ham2=[A -C'*C;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham2);
Ps=x2*inv(x1);L=Ps*C';Ps=real(Ps);L=real(L);
% Rezolvare ecuatie Ricatti 3
Q1=8.1*eye(8,8);R1=.1*eye(2,2);                    % !!! Q1 si R1
ham3=[A -B*pinv(R1)*B'+(miu1^(-2))*G*G';-Q1 -A'];
[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham3);P_inf=x2*inv(x1);K_inf=(pinv(R1))*B'*P_inf;K_inf=real(K_inf)
% Rezolvare ecuatie Ricatti 4
ham4=[A -C'*C+(miu2^(-2))*C1'*C1;-G*G' -A'];[x1,x2,fail,reig_min,epkgdif]=ric_eig(ham4);
Ps_inf=x2*inv(x1);Ps_inf=real(Ps_inf);L_inf=Ps_inf*C'*(inv(D22'*D22))
% Calculul matricelor P_c,Ps_c,K_c,L_c
P_c=(1-k(jj))*P+k(jj)*P_inf;Ps_c=(1-k(jj))*Ps+k(jj)*Ps_inf;P_c=real(P_c);K_c=(pinv(R1))*B'*P_c;L_c=Ps_c*C'*inv(D22'*D22);
% Determinarea comenzii u_b
T=[0 0 0 1 0 0 0 0;0 0 1 0 0 0 0 0;V0*a31 V0*a32 a33 a34 0 0 b31 b32;V0 0 0 0 0 0 0 0;
  V0*a11 V0*a12 0 a14 0 0 b11 b12;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0];
% Calculul matricelor algoritmului
A_c=0.0001*T*A*inv(T);B_c=T*B;
A_c11=A_c(1:5,1:5);A_c12=A_c(1:5,6:8);A_c21=A_c(6:8,1:5);A_c22=A_c(6:8,6:8);B_c1=B_c(1:5,:);B_c2=B_c(6:8,:);
A_c11_p=A_c11-[0 1 0 0 0;0 0 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 0 0];
A_c_ita=A_c22-B_c2*(pinv(B_c1))*A_c12;A_c_csi=A_c21-B_c2*(pinv(B_c1))*A_c11_p;
M=B_c2*pinv(B_c1);B_c_z(:,1)=M(:,3);B_c_z(:,2)=M(:,5);B_c_y=[B_c_z A_c_csi];
sim('Planare_ALS6_sch');
H_k(jj,:)=H(1,:);ALT(jj,:)=H(1,:);Vx_k(jj,:)=Vx(1,:);gama_k(jj,:)=gama(1,:);teta_k(jj,:)=Teta(1,:);tetac_k(jj,:)=tetac(1,:);
disp('--------------------------------------------------------------------------------------------------------------');
end   % end jj
subplot(2,2,1);plot(t,H_k(1,:),'b',t,H_k(2,:),'k',t,H_k(3,:),'r');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(2,2,2);plot(t,(H_ref'-ALT(1,:)),'b',t,(H_ref'-ALT(2,:)),'k',t,(H_ref'-ALT(3,:)),'r');grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
subplot(2,2,3);plot(t,V0*ones(1,length(Vx))-Vx_k(1,:),'b',t,V0*ones(1,length(Vx))-Vx_k(2,:),'k',t,V0*ones(1,length(Vx))-Vx_k(3,:),'r');
```

```
grid;ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(2,2,4);plot(t,gama_c*ones(1,length(gama))-gama_k(1,:),'b',t,gama_c*ones(1,length(gama))-
gama_k(2,:),'k',t,gama_c*ones(1,length(gama))-gama_k(3,:),'r');
grid;ylabel('gama_c-gama [grd');xlabel('Timp [s]');
h=figure;subplot(1,2,1);plot(t,Vx_k(1,:),'b',t,Vx_k(2,:),'k',t,Vx_k(3,:),'r');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(1,2,2);plot(t,gama_k(1,:),'b',t,gama_k(2,:),'k',t,gama_k(3,:),'r');grid;ylabel('gama [grd');xlabel('Timp [s]');
% Caracteristici grafice
sim('Planare_ALS6_sch');
if kk==2
    Vx_er=Vx;Teta_er=Teta;wy_er=wy;gama_er=gama;Alfa_er=Alfa;Delta_p_er=Delta_p;Delta_t_er=Delta_t;
    H_ref_er=H_ref;teta_b_er=teta_b;Vz_er=Vz;H_er=H;Vx_b_er=Vx_b;
end
disp('-------------------------------------------------------------------------------------------------------------');
end
subplot(4,3,1);plot(t,Vx(1,:),t,Vx_er(1,:),'r--');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(4,3,2);plot(t,Vz(1,:),t,Vz_er(1,:),'r--');grid;ylabel('Vz [m/s]');xlabel('Timp [s]');
subplot(4,3,3);plot(t,wy(1,:),t,wy_er(1,:),'r--');grid;ylabel('Omegay [grd/s]');xlabel('Timp [s]');
subplot(4,3,4);plot(t,Teta(1,:),t,Teta_er(1,:),'r--');grid;ylabel('Teta [grd]');xlabel('Timp [s]');
subplot(4,3,5);plot(t,Alfa(1,:),t,Alfa_er(1,:),'r--');grid;ylabel('Alfa [grd]');xlabel('Timp [s]');
subplot(4,3,6);plot(t,gama(1,:),t,gama_er(1,:),'r--');grid;ylabel('gama [grd]');xlabel('Timp [s]');
subplot(4,3,7);plot(t,H(1,:),t,H_er(1,:),'r--');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(4,3,8);plot(t,Delta_p(1,:),t,Delta_p_er(1,:),'r--');grid;ylabel('Bracaj profundor [grd]');xlabel('Timp [s]');
subplot(4,3,9);plot(t,Delta_t(1,:),t,Delta_t_er(1,:),'r--');grid;ylabel('Comanda maneta gaze [grd]');xlabel('Timp [s]');
subplot(4,3,10);plot(t,-Vx(1,:)+V0*ones(1,length(Vx)),t,-Vx_er(1,:)+V0*ones(1,length(Vx)),'r--');
grid;ylabel('Vxc_b-Vx [m]');xlabel('Timp [s]');
subplot(4,3,11); ;ylabel('gama_c-gama [grd'); xlabel('Timp [s]');
plot(t,-gama(1,:)+gama_c*ones(1,length(gama)),t,-gama_er(1,:)+gama_c*ones(1,length(gama)),'r--');grid
subplot(4,3,12);plot(t,(H_ref'-H(1,:)),t,(H_ref_er'-H_er(1,:)),'r--');grid;ylabel('H_ref-H [m]');xlabel('Timp [s]');
```

Appendix 5.1 – Control of aircraft landing during the glide slope phase (longitudinal plane)

```
close all;clear all;
V0=67;Hp=100;H0=3.25;H_ref=0;gama_c=-2.5;x_p=0;Vx_c=V0;Ec0=zeros(5,1);ddd=0.1;M1=-5;M2=5;kphp=0.004;
d0=30;kR=1;x0=0;wr0=2.5;p=.5;yb2=[0;0];yb1=[0;0];yb0=[0.5;70];csi=0.7;b01=-1;b0=10;l21=.01;b02=.3;
l11=0;l01=0;l02=10;l12=0.2;bw=1;Sw=0.5;Sv=0.5;k=20;Deltap0=0;Deltat0=0;T=0.1;Tp=.1;Tt=2.5;Tau=2;
a=[1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1];Zb=10;kz=0.01;kv=0.05;W_0=zeros(11,2);V_0=zeros(10,10);
A=[0 0.122 0 -9.69;0 -0.7535 0.9744 0;0 -0.005 -0.2136 0;0 0 1 0];B=[0 0.3;-0.1667 0;-1.8 0;0 0];
C=eye(4);D=zeros(size(A,1),size(B,2));stare0=[70;0;0.2;-0.45];
a11=A(1,1);a12=A(1,2);a13=A(1,3);a14=A(1,4);a21=A(2,1);a22=A(2,2);a23=A(2,3);a24=A(2,4);
a31=A(3,1);a32=A(3,2);a33=A(3,3);a34=A(3,4);a41=A(4,1);a42=A(4,2);a43=A(4,3);a44=A(4,4);
b11=B(1,1);b12=B(1,2);b21=B(2,1);b22=B(2,2);b31=B(3,1);b32=B(3,2);
c1=a31*a11+a32*a21+a33*a31+b0*a31;c2=a31*a12+a32*a22+a33*a32+b0*a32;
c3=a31*a13+a32*a23+a33*a33+a34+b0*a33;c4=a31*a14+a32*a24+a33*a34+b0*a34;
cp=a31*b11+a32*b21+a33*b31-b31/Tp+b0*b31;ct=a31*b12+a32*b22+a33*b32+b0*b32;
d1=a11*a11+a12*a21+a13*a31;d2=a11*a12+a12*a22+a13*a32;
d3=a12*a23+a13*a33+a14+a11*a13;d4=a11*a14+a12*a24+a13*a34;
dp=a11*b11+a12*b21+a13*b31;dt=a11*b12+a12*b22+a13*b32-b12/Tt;
s1=-.1;s2=-.2;s3=-.3;  A1=[s1 1;s2 1;s3 1];B1=[-s1^3-s1^2;-s2^3-s2^2;-s3^3-s3^2];
x1=lsqr(A1,B1);kd1=x1(1,:);kp1=x1(2,:);r1=roots([1 1 kd1 kp1])
s1=-1;s2=-.2;  A2=[s1 1;s2 1];B2=[-s1^2;-s2^2];
x2=lsqr(A2,B2);kd2=x2(1,:);kp2=x2(2,:);r2=roots([1 kd2 kp2])
Dc=0.1*[kp1 kd1 0 0 0;0 0 0 kp2 kd2]
A_b=[0 1 0 0 0;0 0 1 0 0;-b01*kp1 -b01*kd1 -l21 0 0;0 0 0 0 1;0 0 0 -b02*kp2 -(b02*kd2+l12)];
C_b=[1 0 0 0 0;0 0 0 1 0];R=[-1;-.5;-.3;-.4;-7];
L=place(A_b,C_b',R);L=L';eig(A_b-L*C_b)
Q=.007*eye(5,5);P=lyap(A_b',Q);Bb=[0 0;0 0;b01 0;0 0;0 b02];
MM=[b31/Tp 0;0 b12/Tt];NN=[c2 c3 cp ct;d2 d3 dp dt];
kph=0.01;kih=0.0011;kdh=0.45;
const=2;
sim('F_1');
for i=1:length(Vx)
   Vx_(i,2)=Vx(:,:,i);Teta_(i,2)=Teta(:,:,i);teta_c_(i,2)=teta_c(:,:,i);Alfa_(i,2)=Alfa(:,:,i);gama_(i,2)=gama(:,:,i);wy_(i,2)=wy(:,:,i);
   H_(i,2)=H(:,:,i);vpd1(i,2)=vpd(i,1);vpd2(i,2)=vpd(i,2);vb1(i,2)=vb(1,:,i);vb2(i,2)=vb(2,:,i);va1(i,2)=va(1,:,i);va2(i,2)=va(2,:,i);
   vac1(i,2)=vac(1,:,i);vac2(i,2)=vac(2,:,i);vc1(i,2)=vc(1,:,i);vc2(i,2)=vc(2,:,i);v1(i,2)=v(1,:,i);v2(i,2)=v(2,:,i);
   eps1(i,2)=eps(1,:,i);eps2(i,2)=eps(2,:,i);u1(i,2)=u(1,:,i);u2(i,2)=u(2,:,i);x_(i,2)=x(:,:,i);vh1_(i,2)=vh1(i,1);
end
const=1;
sim('F_1');
for i=1:length(Vx)
```

```
    Vx_(i,1)=Vx(:,:,i);Teta_(i,1)=Teta(:,:,i);teta_c_(i,1)=teta_c(:,:,i);Alfa_(i,1)=Alfa(:,:,i);gama_(i,1)=gama(:,:,i);wy_(i,1)=wy(:,:,i);
    H_(i,1)=H(:,:,i);vpd1(i,1)=vpd(i,1);vpd2(i,1)=vpd(i,2);vb1(i,1)=vb(1,:,i);vb2(i,1)=vb(2,:,i);va1(i,1)=va(1,:,i);va2(i,1)=va(2,:,i);
    vac1(i,1)=vac(1,:,i);vac2(i,1)=vac(2,:,i);vc1(i,1)=vc(1,:,i);vc2(i,1)=vc(2,:,i);v1(i,1)=v(1,:,i);v2(i,1)=v(2,:,i);
    eps1(i,1)=eps(1,:,i);eps2(i,1)=eps(2,:,i);u1(i,1)=u(1,:,i);u2(i,1)=u(2,:,i);x_(i,1)=x(:,:,i);vh1_(i,1)=vh1(i,1);
end
subplot(6,4,1);plot(t,Vx_(:,1),'b',t,Vx_(:,2),'r--');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(6,4,2);plot(t,Teta_(:,1),'b',t,Teta_(:,2),'r--');grid;ylabel('Teta [grd]');xlabel('Timp [s]');
subplot(6,4,3);plot(t,teta_c_(:,1),'b',t,teta_c_(:,2),'r--');grid;ylabel('teta_c [grd]');xlabel('Timp [s]');
subplot(6,4,4);plot(t,Alfa_(:,1),'b',t,Alfa_(:,2),'r');grid;ylabel('alfa [grd]');xlabel('Timp [s]');
subplot(6,4,5);plot(t,gama_(:,1),'b',t,gama_(:,2),'r--');grid;ylabel('Gama [grd]');xlabel('Timp [s]');
subplot(6,4,6);plot(t,wy_(:,1),'b',t,wy_(:,2),'r--');grid;ylabel('wy [grd/s]');xlabel('Timp [s]');
subplot(6,4,7);plot(t,H_(:,1),'b',t,H_(:,2),'r--');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(6,4,8);plot(t,vh1_(:,1),'b',t,vh1_(:,2),'r--');grid;ylabel('vh [??]');xlabel('Timp [s]');
subplot(6,4,9);plot(t,vpd1(:,1),'b',t,vpd1(:,2),'r--');grid;ylabel('vpd1');xlabel('Timp [s]');
subplot(6,4,10);plot(t,vpd2(:,1),'b',t,vpd2(:,2),'r--');grid;ylabel('vpd2');xlabel('Timp [s]');
subplot(6,4,11);plot(t,vb1(:,1),'b',t,vb1(:,2),'r--');grid;ylabel('vb1');xlabel('Timp [s]');
subplot(6,4,12);plot(t,vb2(:,1),'b',t,vb2(:,2),'r--');grid;ylabel('vb2');xlabel('Timp [s]');
subplot(6,4,13);plot(t,va1(:,1),'b',t,va1(:,2),'r--');grid;ylabel('va1');xlabel('Timp [s]');
subplot(6,4,14);plot(t,va2(:,1),'b',t,va2(:,2),'r--');grid;ylabel('va1');xlabel('Timp [s]');
subplot(6,4,15);plot(t,vac1(:,1),'b',t,vac1(:,2),'r--');grid;ylabel('vac1');xlabel('Timp [s]');
subplot(6,4,16);plot(t,vac2(:,1),'b',t,vac2(:,2),'r--');grid;ylabel('vac2');xlabel('Timp [s]');
subplot(6,4,17);plot(t,vc1(:,1),'b',t,vc1(:,2),'r--');grid;ylabel('vc1');xlabel('Timp [s]');
subplot(6,4,18);plot(t,vc2(:,1),'b',t,vc2(:,2),'r--');grid;ylabel('vc2');xlabel('Timp [s]');
subplot(6,4,19);plot(t,v1(:,1),'b',t,v1(:,2),'r--');grid;ylabel('v1');xlabel('Timp [s]');
subplot(6,4,20);plot(t,v2(:,1),'b',t,v2(:,2),'r--');grid;ylabel('v2');xlabel('Timp [s]');
subplot(6,4,21);plot(t,eps1(:,1),'b',t,eps1(:,2),'r--');grid;ylabel('eps1');xlabel('Timp [s]');
subplot(6,4,22);plot(t,eps2(:,1),'b',t,eps2(:,2),'r--');grid;ylabel('eps2');xlabel('Timp [s]');
subplot(6,4,23);plot(t,u1(:,1),'b',t,u1(:,2),'r--');grid;ylabel('u1');xlabel('Timp [s]');
subplot(6,4,24);plot(t,u2(:,1),'b',t,u2(:,2),'r--');grid;ylabel('u2');xlabel('Timp [s]');
h=figure;plot(Teta_(:,1),wy_(:,1),'b',Teta_(:,2),wy_(:,2),'r--');grid;xlabel('Teta');ylabel('Teta derivat');
```

Appendix 5.2 – Control of aircraft landing during the flare phase (longitudinal plane)

```
close all;clear all;
V0=67;Hp=100;H0=3.25;H_ref=0;gama_c=-2.5;x_p=0;Vx_c=V0;Ec0=zeros(5,1);ddd=0.1;M1=-5;M2=5;kphp=0.004;
d0=30;kR=1;x0=50;wr0=2.5;p=.5;yb2=[0;0];yb1=[0;0];yb0=[0;0];csi=0.7;b01=-1;b0=10;l21=.01;b02=.3;
l11=0;l01=0;l02=10;l12=0.2;bw=1;Sw=0.5;Sv=0.5;k=20;Deltap0=0;Deltat0=0;T=0.1;Tp=.1;Tt=2.5;Tau=2;
a=[1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1];Zb=10;kz=0.01;kv=0.05;W_0=zeros(11,2);V_0=zeros(10,10);
A=[0 0.122 0 -9.69;0 -0.7535 0.9744 0;0 -0.005 -0.2136 0;0 0 1 0];B=[0 0.3;-0.1667 0;-1.8 0;0 0];
C=eye(4);D=zeros(size(A,1),size(B,2));stare0=[67;0;0;0];
a11=A(1,1);a12=A(1,2);a13=A(1,3);a14=A(1,4);a21=A(2,1);a22=A(2,2);a23=A(2,3);a24=A(2,4);
a31=A(3,1);a32=A(3,2);a33=A(3,3);a34=A(3,4);a41=A(4,1);a42=A(4,2);a43=A(4,3);a44=A(4,4);
b11=B(1,1);b12=B(1,2);b21=B(2,1);b22=B(2,2);b31=B(3,1);b32=B(3,2);
c1=a31*a11+a32*a21+a33*a31+b0*a31;c2=a31*a12+a32*a22+a33*a32+b0*a32;
c3=a31*a13+a32*a23+a33*a33+a34+b0*a33;c4=a31*a14+a32*a24+a33*a34+b0*a34;
cp=a31*b11+a32*b21+a33*b31-b31/Tp+b0*b31;ct=a31*b12+a32*b22+a33*b32+b0*b32;
d1=a11*a11+a12*a21+a13*a31;d2=a11*a12+a12*a22+a13*a32;
d3=a12*a23+a13*a33+a14+a11*a13;d4=a11*a14+a12*a24+a13*a34;
dp=a11*b11+a12*b21+a13*b31;dt=a11*b12+a12*b22+a13*b32-b12/Tt;
s1=-.1;s2=-.2;s3=-.3;   A1=[s1 1;s2 1;s3 1];B1=[-s1^3-s1^2;-s2^3-s2^2;-s3^3-s3^2];
x1=lsqr(A1,B1);kd1=x1(1,:);kp1=x1(2,:);r1=roots([1 1 kd1 kp1])
s1=-1;s2=-.2;  A2=[s1 1;s2 1];B2=[-s1^2;-s2^2];
x2=lsqr(A2,B2);kd2=x2(1,:);kp2=x2(2,:);r2=roots([1 kd2 kp2])
Dc=0.1*[kp1 kd1 0 0 0;0 0 0 kp2 kd2]
A_b=[0 1 0 0 0;0 0 1 0 0;-b01*kp1 -b01*kd1 -l21 0 0;0 0 0 0 1;0 0 0 -b02*kp2 -(b02*kd2+l12)];
C_b=[1 0 0 0 0;0 0 0 1 0];R=[-1;-.5;-.3;-.4;-7];
L=place(A_b,C_b',R);L=L';eig(A_b-L*C_b)
Q=.007*eye(5,5);P=lyap(A_b',Q);Bb=[0 0;0 0;b01 0;0 0;0 b02];
MM=[b31/Tp 0;0 b12/Tt];NN=[c2 c3 cp ct;d2 d3 dp dt];kph=0.01;kih=0.0011;kdh=0.45; const=2;
sim('F_2');
for i=1:length(Vx)
    Vx_(i,2)=Vx(:,:,i);Teta_(i,2)=Teta(:,:,i);Alfa_(i,2)=Alfa(:,:,i);wy_(i,2)=wy(:,:,i);gama_(i,2)=gama(:,:,i);
    H_(i,2)=H(:,:,i);vpd1(i,2)=vpd(i,1);vpd2(i,2)=vpd(i,2);vb1(i,2)=vb(1,:,i);vb2(i,2)=vb(2,:,i);
    va1(i,2)=va(1,:,i);va2(i,2)=va(2,:,i);vac1(i,2)=vac(1,:,i);vac2(i,2)=vac(2,:,i);vc1(i,2)=vc(1,:,i);vc2(i,2)=vc(2,:,i);
    v1(i,2)=v(1,:,i);v2(i,2)=v(2,:,i);eps1(i,2)=eps(1,:,i);eps2(i,2)=eps(2,:,i);u1(i,2)=u(1,:,i);u2(i,2)=u(2,:,i);
    x1(i,2)=x(:,:,i);vh1_(i,2)=vh1(i,1);Hd_(i,2)=Hd(:,:,i);
end
const=1;
sim('F_2');
for i=1:length(Vx)
```

```
   Vx_(i,1)=Vx(:,:,i);Teta_(i,1)=Teta(:,:,i);Alfa_(i,1)=Alfa(:,:,i);wy_(i,1)=wy(:,:,i);gama_(i,1)=gama(:,:,i);
   H_(i,1)=H(:,:,i);vpd1(i,1)=vpd(i,1);vpd2(i,1)=vpd(i,2);vb1(i,1)=vb(1,:,i);vb2(i,1)=vb(2,:,i);
   va1(i,1)=va(1,:,i);va2(i,1)=va(2,:,i);vac1(i,1)=vac(1,:,i);vac2(i,1)=vac(2,:,i);vc1(i,1)=vc(1,:,i);vc2(i,1)=vc(2,:,i);
   v1(i,1)=v(1,:,i);v2(i,1)=v(2,:,i);eps1(i,1)=eps(1,:,i);eps2(i,1)=eps(2,:,i);u1(i,1)=u(1,:,i);u2(i,1)=u(2,:,i);
   x1(i,1)=x(:,:,i);vh1_(i,1)=vh1(i,1);Hd_(i,1)=Hd(:,:,i);
end
subplot(6,4,1);plot(t,Vx_(:,1),'b',t,Vx_(:,2),'r--');grid;ylabel('Vx [m/s]');xlabel('Timp [s]');
subplot(6,4,2);plot(t,Teta_(:,1),'b',t,Teta_(:,2),'r--');grid;ylabel('Teta [grd]');xlabel('Timp [s]');
subplot(6,4,3);plot(t,Alfa_(:,1),'b',t,Alfa_(:,2),'r');grid;ylabel('alfa [grd]');xlabel('Timp [s]');
subplot(6,4,4);plot(t,gama_(:,1),'b',t,gama_(:,2),'r--');grid;ylabel('Gama [grd]');xlabel('Timp [s]');
subplot(6,4,5);plot(t,wy_(:,1),'b',t,wy_(:,2),'r--');grid;ylabel('wy [grd/s]');xlabel('Timp [s]');
subplot(6,4,6);plot(t,Hd_(:,1),'b',t,Hd_(:,2),'r--');grid;ylabel('Vit. ascens [m/s]');xlabel('Timp [s]');
subplot(6,4,7);plot(t,H_(:,1),'b',t,H_(:,2),'r--');grid;ylabel('H [m]');xlabel('Timp [s]');
subplot(6,4,8);plot(t,vh1_(:,1),'b',t,vh1_(:,2),'r--');grid;ylabel('vh [??]');xlabel('Timp [s]');
subplot(6,4,9);plot(t,vpd1(:,1),'b',t,vpd1(:,2),'r--');grid;ylabel('vpd1');xlabel('Timp [s]');
subplot(6,4,10);plot(t,vpd2(:,1),'b',t,vpd2(:,2),'r--');grid;ylabel('vpd2');xlabel('Timp [s]');
subplot(6,4,11);plot(t,vb1(:,1),'b',t,vb1(:,2),'r--');grid;ylabel('vb1');xlabel('Timp [s]');
subplot(6,4,12);plot(t,vb2(:,1),'b',t,vb2(:,2),'r--');grid;ylabel('vb2');xlabel('Timp [s]');
subplot(6,4,13);plot(t,va1(:,1),'b',t,va1(:,2),'r--');grid;ylabel('va1');xlabel('Timp [s]');
subplot(6,4,14);plot(t,va2(:,1),'b',t,va2(:,2),'r--');grid;ylabel('va2');xlabel('Timp [s]');
subplot(6,4,15);plot(t,vac1(:,1),'b',t,vac1(:,2),'r--');grid;ylabel('vac1');xlabel('Timp [s]');
subplot(6,4,16);plot(t,vac2(:,1),'b',t,vac2(:,2),'r--');grid;ylabel('vac2');xlabel('Timp [s]');
subplot(6,4,17);plot(t,vc1(:,1),'b',t,vc1(:,2),'r--');grid;ylabel('vc1');xlabel('Timp [s]');
subplot(6,4,18);plot(t,vc2(:,1),'b',t,vc2(:,2),'r--');grid;ylabel('vc2');xlabel('Timp [s]');
subplot(6,4,19);plot(t,v1(:,1),'b',t,v1(:,2),'r--');grid;ylabel('v1');xlabel('Timp [s]');
subplot(6,4,20);plot(t,v2(:,1),'b',t,v2(:,2),'r--');grid;ylabel('v2');xlabel('Timp [s]');
subplot(6,4,21);plot(t,eps1(:,1),'b',t,eps1(:,2),'r--');grid;ylabel('eps1');xlabel('Timp [s]');
subplot(6,4,22);plot(t,eps2(:,1),'b',t,eps2(:,2),'r--');grid;ylabel('eps2');xlabel('Timp [s]');
subplot(6,4,23);plot(t,u1(:,1),'b',t,u1(:,2),'r--');grid;ylabel('u1');xlabel('Timp [s]');
subplot(6,4,24);plot(t,u2(:,1),'b',t,u2(:,2),'r--');grid;ylabel('u2');xlabel('Timp [s]');
h=figure;plot(Teta_(:,1),wy_(:,1),'b',Teta_(:,2),wy_(:,2),'r--');grid;xlabel('Teta');ylabel('Teta derivat');
```

Appendix 5.3 – Control of aircraft landing (lateral-directional plane)

```
close all; clear all;
for const=1:2
y2c=0;wr0=0.9;p=0.5;b0=1;yb2=[0;0];yb1=[0;0];yb0=[-1;1];csi=0.7;
Ec0=zeros(6,1);Te=0.4;Td=0.4;Deltae0=30;Deltad0=30;V0=67;R0=6700;
ddd=0.1;M1=-5;M2=5;T=0.01;kcsi=0.5;k_L=1;b01=b0;b02=b0;L12=b0;L21=b0;
kc=1;Ti=1000;T_d=0.1;kp=kc;ki=kc/Ti;kd=kc*T_d;
a=[1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1];Zb=10;kz=0.01;kv=0.01;Sw=0.05;Sv=0.5;k=20;bw=1;W_0=zeros(11,2);V_0=zeros(10,10);
A=[-0.089 0 -67 9.81 0;-1.33 -0.98 0.33 0 0;0.17 -0.17 -0.217 0 0;0 1 0 0 0;0 0 1 0 0];
B=[0 1;0.23 0.06;0.026 -0.15;0 0;0 0];C=eye(5);D=zeros(size(A,1),size(B,2));stare0=10*[.4;-0.1;0.1;0.1;0.1];
a11=A(1,1);a12=A(1,2);a13=A(1,3);a14=A(1,4);a15=A(1,5);a21=A(2,1);a22=A(2,2);a23=A(2,3);a24=A(2,4);a25=A(2,5);
a31=A(3,1);a32=A(3,2);a33=A(3,3);a34=A(3,4);a35=A(3,5);a41=A(4,1);a42=A(4,2);a43=A(4,3);a44=A(4,4);a45=A(4,5);
a51=A(5,1);a52=A(5,2);a53=A(5,3);a54=A(5,4);a55=A(5,5);b11=B(1,1);b12=B(1,2);b21=B(2,1);b22=B(2,2);
b31=B(3,1);b32=B(3,2);b41=B(4,1);b42=B(4,2);b51=B(5,1);b52=B(5,2);
c1=a21*a11+a22*a21+a23*a31+b0*a21;c2=a22^2+a23*a32+b0*a22;c3=a21*a13+a22*a23+a23*a33+b0*a23;
c4=a21*a14;ce=a22*b21+a23*b31-b21/Te+b0*b21;cd=a21*b12+a22*b22+a23*b32-b22/Td+b0*b22;
d1=a31*a11+a32*a21+a33*a31+b0*a31;d2=a32*a22+a33*a32+b0*a32;d3=a31*a13+a32*a23+a33^2+b0*a33;
d4=a31*a14;de=a32*b21+a33*b31-b31/Te+b0*b31;dd=a31*b12+a32*b22+a33*b32-b32/Td+b0*b32;
MM=[b21/Te b22/Td;b31/Te b32/Td];NN=[c1 c2 c3 ce cd;d1 d2 d3 de dd];
s1=-0.1;s2=-0.2;s3=-0.3;
A1=b01*[s1 1;s2 1;s3 1];B1=[-(s1^3)-L21*(s1^2);-(s2^3)-L21*(s2^2);-(s3^3)-L21*(s3^2)];
x1=lsqr(A1,B1);kd1=x1(1,:);kp1=x1(2,:);rad_1=roots([1 L21 b01*kd1 b01*kp1])
s1=-1.1;s2=-0.7;     A2=b02*[s1 1;s2 1];B2=[-s1^2-L12*s1;-s2^2-L12*s2];
x2=lsqr(A2,B2);kd2=x2(1,:);kp2=x2(2,:);rad_2=roots([1 (b02*kd2+L12) (b02*kp2)])
Dc=[kp1 kd1 0 0 0 0;0 0 0 kp2 kd2 0];AA=[0 1 0 0 0 0;0 0 1 0 0 0;0 0 -L21 0 0 0;0 0 0 0 1 0;0 0 0 0 0 1;0 0 0 0 0 -L12];
BB=[0 0;0 0;1 0;0 0;0 0;0 1];A_b=AA-BB*Dc;B_b=BB;C_b=[1 0 0 0 0 0;0 0 0 1 0 0];
R=[-1;-0.5;-0.3;-6;-7;-0.5];L=place(A_b,C_b',R);L=L';A_t=A_b-L*C_b;eig(A_t)
Q=0.01*eye(6,6);P=lyap(A_b',Q);
if const==1
  sim('Lat_sch');
  for i=1:length(Vy)
     Vy_1(i,:)=Vy(:,:,i);wx_1(i,:)=wx(:,:,i);wz_1(i,:)=wz(:,:,i);fi_1(i,:)=fi(:,:,i);d_psi_1(i,:)=d_psi(:,:,i);
     Y_1(i,:)=Y(:,:,i);Lambda_1(i,:)=Lambda(:,:,i);delta_e_1(i,:)=delta_e(i,:);delta_d_1(i,:)=delta_d(i,:);
     u1_1(i,:)=u(1,:,i);u2_1(i,:)=u(2,:,i);v1_1(i,:)=v(1,:,i);v2_1(i,:)=v(2,:,i);
     ep1_1(i,:)=ep(1,:,i);ep2_1(i,:)=ep(2,:,i);vc1_1(i,:)=vc(1,:,i);vc2_1(i,:)=vc(2,:,i);
     vac1_1(i,:)=vac(1,:,i);vac2_1(i,:)=vac(2,:,i);vpd1_1(i,:)=vpd(i,1);vpd2_1(i,:)=vpd(i,2);
     va1_1(i,:)=va(1,:,i);va2_1(i,:)=va(2,:,i);vb1_1(i,:)=vb(1,:,i);vb2_1(i,:)=vb(2,:,i);
```

```
        vr1_1(i,:)=vr(1,:,i);vr2_1(i,:)=vr(2,:,i);vh1_1(i,:)=vh(i,1);vh2_1(i,:)=vh(i,2);
    end
else
    sim('Lat_sch');
    for i=1:length(Vy)
        Vy_2(i,:)=Vy(:,:,i);wx_2(i,:)=wx(:,:,i);wz_2(i,:)=wz(:,:,i);fi_2(i,:)=fi(:,:,i);d_psi_2(i,:)=d_psi(:,:,i);
        Y_2(i,:)=Y(:,:,i);Lambda_2(i,:)=Lambda(:,:,i);delta_e_2(i,:)=delta_e(i,:);delta_d_2(i,:)=delta_d(i,:);
        u1_2(i,:)=u(1,:,i);u2_2(i,:)=u(2,:,i);v1_2(i,:)=v(1,:,i);v2_2(i,:)=v(2,:,i);
        ep1_2(i,:)=ep(1,:,i);ep2_2(i,:)=ep(2,:,i);vc1_2(i,:)=vc(1,:,i);vc2_2(i,:)=vc(2,:,i);
        vac1_2(i,:)=vac(1,:,i);vac2_2(i,:)=vac(2,:,i);vpd1_2(i,:)=vpd(i,1);vpd2_2(i,:)=vpd(i,2);
        va1_2(i,:)=va(1,:,i);va2_2(i,:)=va(2,:,i);vb1_2(i,:)=vb(1,:,i);vb2_2(i,:)=vb(2,:,i);
        vr1_2(i,:)=vr(1,:,i);vr2_2(i,:)=vr(2,:,i);vh1_2=zeros(length(Vy),1);vh2_2=zeros(length(Vy),1);
    end
end
end
figure(1);
subplot(9,3,1);plot(t,Vy_1);grid;ylabel('Vy [m/s]');xlabel('Timp [s]');
subplot(9,3,2);plot(t,wx_1);grid;ylabel('wx [grd/s]');xlabel('Timp [s]');
subplot(9,3,3);plot(t,wz_1);grid;ylabel('wz [grd/s]');xlabel('Timp [s]');
subplot(9,3,4);plot(t,fi_1);grid;ylabel('fi [grd]');xlabel('Timp [s]');
subplot(9,3,5);plot(t,d_psi_1);grid;ylabel('d_psi [grd]');xlabel('Timp [s]');
subplot(9,3,6);plot(t,u1_1);grid;ylabel('u1 [grd]');xlabel('Timp [s]');
subplot(9,3,7);plot(t,u2_1);grid;ylabel('u2 [grd]');xlabel('Timp [s]');
subplot(9,3,8);plot(t,Y_1);grid;ylabel('Y [m]');xlabel('Timp [s]');
subplot(9,3,9);plot(t,Lambda_1);grid;ylabel('Lambda [grd]');xlabel('Timp [s]');
subplot(9,3,10);plot(t,v1_1);grid;ylabel('v1 [grd]');xlabel('Timp [s]');
subplot(9,3,11);plot(t,v2_1);grid;ylabel('v2 [grd]');xlabel('Timp [s]');
subplot(9,3,12);plot(t,ep1_1);grid;ylabel('eps1 [grd]');xlabel('Timp [s]');
subplot(9,3,13);plot(t,ep2_1);grid;ylabel('eps2 [grd]');xlabel('Timp [s]');
subplot(9,3,14);plot(t,vc1_1);grid;ylabel('vc1 [grd]');xlabel('Timp [s]');
subplot(9,3,15);plot(t,vc2_1);grid;ylabel('vc2 [grd]');xlabel('Timp [s]');
subplot(9,3,16);plot(t,vac1_1);grid;ylabel('vac1 [grd]');xlabel('Timp [s]');
subplot(9,3,17);plot(t,vac2_1);grid;ylabel('vac2 [grd]');xlabel('Timp [s]');
subplot(9,3,18);plot(t,vpd1_1);grid;ylabel('vpd1 [grd]');xlabel('Timp [s]');
subplot(9,3,19);plot(t,vpd2_1);grid;ylabel('vpd2 [grd]');xlabel('Timp [s]');
subplot(9,3,20);plot(t,va1_1);grid;ylabel('va1 [grd]');xlabel('Timp [s]');
subplot(9,3,21);plot(t,va2_1);grid;ylabel('va2 [grd]');xlabel('Timp [s]');
subplot(9,3,22);plot(t,vb1_1);grid;ylabel('vb1 [grd]');xlabel('Timp [s]');
subplot(9,3,23);plot(t,vb2_1);grid;ylabel('vb2 [grd]');xlabel('Timp [s]');
subplot(9,3,24);plot(t,delta_e_1);grid;ylabel('delta_e [grd]');xlabel('Timp [s]');
subplot(9,3,25);plot(t,delta_d_1);grid;ylabel('delta_d [grd]');xlabel('Timp [s]');
subplot(9,3,26);plot(t,vh1_1);grid;ylabel('vh1 [grd]');xlabel('Timp [s]');
subplot(9,3,27);plot(t,vh2_1);grid;ylabel('vh2 [grd]');xlabel('Timp [s]');
figure(2);
subplot(9,3,1);plot(t,Vy_2);grid;ylabel('Vy [m/s]');xlabel('Timp [s]');
subplot(9,3,2);plot(t,wx_2);grid;ylabel('wx [grd/s]');xlabel('Timp [s]');
subplot(9,3,3);plot(t,wz_2);grid;ylabel('wz [grd/s]');xlabel('Timp [s]');
subplot(9,3,4);plot(t,fi_2);grid;ylabel('fi [grd]');xlabel('Timp [s]');
subplot(9,3,5);plot(t,d_psi_2);grid;ylabel('d_psi [grd]');xlabel('Timp [s]');
subplot(9,3,6);plot(t,delta_e_2);grid;ylabel('delta_e [grd]');xlabel('Timp [s]');
subplot(9,3,7);plot(t,delta_d_2);grid;ylabel('delta_d [grd]');xlabel('Timp [s]');
subplot(9,3,8);plot(t,u1_2);grid;ylabel('u1 [grd]');xlabel('Timp [s]');
subplot(9,3,9);plot(t,u2_2);grid;ylabel('u2 [grd]');xlabel('Timp [s]');
subplot(9,3,10);plot(t,Y_2);grid;ylabel('Y [m]');xlabel('Timp [s]');
subplot(9,3,11);plot(t,Lambda_2);grid;ylabel('Lambda [grd]');xlabel('Timp [s]');
subplot(9,3,12);plot(t,v1_2);grid;ylabel('v1 [grd]');xlabel('Timp [s]');
subplot(9,3,13);plot(t,v2_2);grid;ylabel('v2 [grd]');xlabel('Timp [s]');
subplot(9,3,14);plot(t,ep1_2);grid;ylabel('eps1 [grd]');xlabel('Timp [s]');
subplot(9,3,15);plot(t,ep2_2);grid;ylabel('eps2 [grd]');xlabel('Timp [s]');
subplot(9,3,16);plot(t,vc1_2);grid;ylabel('vc1 [grd]');xlabel('Timp [s]');
subplot(9,3,17);plot(t,vc2_2);grid;ylabel('vc2 [grd]');xlabel('Timp [s]');
subplot(9,3,18);plot(t,vac1_2);grid;ylabel('vac1 [grd]');xlabel('Timp [s]');
subplot(9,3,19);plot(t,vac2_2);grid;ylabel('vac2 [grd]');xlabel('Timp [s]');
subplot(9,3,20);plot(t,vpd1_2);grid;ylabel('vpd1 [grd]');xlabel('Timp [s]');
subplot(9,3,21);plot(t,vpd2_2);grid;ylabel('vpd2 [grd]');xlabel('Timp [s]');
subplot(9,3,22);plot(t,va1_2);grid;ylabel('va1 [grd]');xlabel('Timp [s]');
subplot(9,3,23);plot(t,va2_2);grid;ylabel('va2 [grd]');xlabel('Timp [s]');
subplot(9,3,24);plot(t,vb1_2);grid;ylabel('vb1 [grd]');xlabel('Timp [s]');
subplot(9,3,25);plot(t,vb2_2);grid;ylabel('vb2 [grd]');xlabel('Timp [s]');
subplot(9,3,26);plot(t,vr1_2);grid;ylabel('vr1 [grd]');xlabel('Timp [s]');
subplot(9,3,27);plot(t,vr2_2);grid;ylabel('vr2 [grd]');xlabel('Timp [s]');
```