# A Fault Detection and Localization of An Helicopter Execution Element

Octavian Grigore
Department of Aerospace Engineering, University “Politehnica” Bucharest-Romania,
+40.21.402.38.69, +40.21.212.92.20, goctav@oberth.avionics.pub.ro

*Abstract* – **Before a system can attempt to deal with a fault, one must first make an assessment of its location. Increasing of an exception indicates the presence of a fault, and in many systems no further information is available to assist fault location. In this case, automatic repair of the system will only be possible if the exception provides an accurate guide to the location of the fault. Like an example a fuzzy FDI system for an execution element of a helicopter is illustrated.**

*Index Terms* – **identification, fault detection, fuzzy control, inference, aircraft control, lqr method.**

## I. Fault Detection And Localization

A system state is erroneous if that state could lead to a failure of the system. The errors consist of those parts of the state that would have to change to prevent the failure from occurring. After an error has been detected and the damage assessment phase has produced an estimate of the extent to which the system state is erroneous, it will be necessary to eliminate those errors from the system state. Thus the system can return to normal operation since the immediate danger of failure has been averted.

It is important to specify that these errors usually are the results of either a design (referred in literature like design fault) or a component failure (referred like component fault). So, practically when it comes to discuss of errors recovery it means the same thing with faults recovery. The techniques that attempt to eradicate faults from a system provide treatment for the fault itself and can be divided into two stages: fault location and system repair.

Thus the mathematical model of such a fault detection system is made by comparing on line the real model $\mathbf{M_R}$ with the nominal one $\mathbf{M_N}$, model that is without faults. This supervised algorithm is represented in Fig. 1.

The $\boldsymbol{u}_{\mathbf{c}}\boldsymbol{(t)}$ vector is the input vector (the command vector), $\boldsymbol{u}_{\mathbf{d}}\boldsymbol{(t)}$ vector is the fault vector of the execution element, $\boldsymbol{u}_{\mathbf{r}}\boldsymbol{(t)}$ vector is the real input vector for the real model, $\boldsymbol{x(t)}$ vector is the output vector, which represents the measured values of the real model $\mathbf{M_R}$, $\boldsymbol{x^*(t)}$ vector is the output vector of the nominal model $\mathbf{M_N}$, $\boldsymbol{r(t)}$ is the residual vector between the outputs of real system $\mathbf{M_R}$ and the nominal $\mathbf{M_N}$.

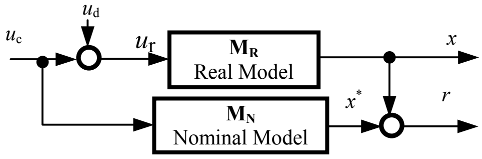


Fig. 1. Scheme used for detection and localization of the fault execution elements

As a model is always an approximation of a real physical phenomenon it means that the precision of this approximation would affect the precision of faults detection and identification. Behind these errors, called *model’s errors*, appear the uncertainties of the environment in which the system is operated. Moreover, the measurement is always affected by noise or systematic errors due to the operating equipment. Let it be a general system, described by [1]:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) + \mathbf{E}_x\mathbf{f}(t) + \mathbf{F}_x\mathbf{n}(t) + \mathbf{G}_x\mathbf{m}(t) \\ \mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Du}(t) + \mathbf{E}_y\mathbf{f}(t) + \mathbf{F}_y\mathbf{n}(t) + \mathbf{G}_y\mathbf{m}(t) \end{cases} \tag{1}$$

where **A**, **B**, **C** and **D** are the known notations for a system written in the state equations. The faults are represented by the error vector $\boldsymbol{f(t)}$, unknown, $\boldsymbol{n(t)}$ is the external uncertain vector and $\boldsymbol{m(t)}$ is the model’s error vector. $\mathbf{E}_x$, $\mathbf{E}_y$, $\mathbf{F}_x$, $\mathbf{F}_y$, $\mathbf{G}_x$ and $\mathbf{G}_y$ are the distribution matrix. If $\mathbf{F}_x$ and $\mathbf{F}_y$ are known then the uncertainties $\boldsymbol{n(t)}$ are called *structured*, otherwise they will be called *unstructured*. Also, if the distribution matrix $\mathbf{G}_x$ and $\mathbf{G}_y$ is known then the model’s errors are structured. In this case the vectors $\boldsymbol{n(t)}$ and $\boldsymbol{m(t)}$ could be considered like unknown inputs.

For difference generation block development, which means to find the residual vector ***r(t)***, it must be fulfilled the following demands:

- maximum sensitiveness for ***r(t)*** in comparison with the changing of the fault vector ***f(t)***, regarding as a input value;
- robustness regarding the model's error ***m(t)***;
- insurance the uncoupling regarding the external uncertainties ***n(t)***.

Starting from the ***r(t)*** signal the difference evaluation block must be able to generate the alarm signals and also to ensure the fault detection and separation.

It is well known that this point of view of residual generation is based on two concepts: the estimation of the state and the identification of the parameters.

The second method that use the on-line identification procedure of the system's parameter [4] [6] it was used. In this case the residual vector ***r(t)*** contains eventual the difference between the physical parameters, the fault detection and the identification are practically simultaneous operations. For the development of such a method it must be proceed to the following steps:

*1)* choose the structure of the parametrical model of the supervised system based on the linear input – output equations;
*2)* specify the nominal values of the $\theta_i$ parameters (standard nominal), which corresponds to a working period without faults (the nominal model $\mathbf{M_N}$);
*3)* start from the input – output experimental values determined, which must be periodically brought them up – to – date to the values of parameters $p_i$, belonging to the model $\mathbf{M_R}$;
*4)* calculate the residual vector ***r(t)***= $[x^*_i - x_i]_{i=1,..,m}$;
*5)* choose the adequate decision to the fault presence and localization.

The advantages of this method are:

- data acquisition will be easily validated using common statistic and numerical methods;
- it doesn't require a computational effort and the used memory is a normal standard one from a PC.

### 1.1 *The Analytic Detection and Localization of a Fault Execution.*

If a fault appeared at a servomechanism's level, the detection and localization of such a fault execution element could be done with the scheme represented in Fig. 1, where: **S** is the real system, **M** is mathematical model of the system S, $\boldsymbol{u}_\mathbf{c} = [u_{c1}\ldots u_{cm}]^T$ is the command vector, $\boldsymbol{u}_\mathbf{d} = [u_{d1}\ldots u_{dm}]^T$ is the error vector (when $u_{dj}= 0$ it can be said that the "j-th" servomechanism is in a well working and if $u_{dj} \neq 0$ the "j-th" servomechanism is in fault) and $\boldsymbol{u}_\mathbf{r} = [u_{r1}\ldots\ldots u_{rm}]^T$ is the real command vector.

When the system is working well, then $u_{ri} = u_{ci}$ $(\forall)$ $i$ =1,..,m.

If it is used the known notations $\boldsymbol{x} = [x_1\ldots x_n]^T$ for the state vector for the **S** system; $\boldsymbol{x}^* = [x_1^*\ldots x_n^*]^T$ the state vector for the mathematical model **M** and $\boldsymbol{r} = [r_1\ldots r_n]^T$ the residual vector, for each component it can be written:

$$r_i = x_i - x_i^*,\ (\forall)\ i=1,\ldots n \tag{2}$$

with the condition of well working $r = 0$.

If the system **S** is:

$$\boldsymbol{S}:\ \ \dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{Ax}(\boldsymbol{t}) + \boldsymbol{B}[\boldsymbol{u_c}(\boldsymbol{t}) + \boldsymbol{u_d}(\boldsymbol{t})] \tag{3}$$

and the associate mathematical model is:

$$\boldsymbol{M}:\ \ \dot{\boldsymbol{x}}^*(t) = \boldsymbol{Ax}^*(t) + \boldsymbol{Bu_c}(t), \tag{4}$$

then the residual vector is:

$$\mathbf{r}(t) = \mathbf{x}(t) - \mathbf{x}^*(t) \tag{5}$$

From (5) it can see that the equation of residual vector is:

$$\dot{\boldsymbol{r}}(\boldsymbol{t}) = \boldsymbol{Ar}(\boldsymbol{t}) + \boldsymbol{Bu_d}(\boldsymbol{t}). \tag{6}$$

where the matrix **A** and **B** are known, and also the residual vector ***r***(*t*).

Practically a well-known scheme for obtaining the error vector deduced from an analytical relation like:

$$\boldsymbol{u_d}(\boldsymbol{t}) = \boldsymbol{f}(\boldsymbol{r}(\boldsymbol{t}), \dot{\boldsymbol{r}}(\boldsymbol{t})) \tag{7}$$

is shown in Fig. 2.

### 1.2 *The Analytic Detection and Localization of a Fault Using Fuzzy Controllers.*

Because solving the equation (7) implies to calculate the differential of the residual vector ***r***(*t*), and also to verify the condition of existence of inverse matrix, for simplification to solve the structure from Fig. 2 by means of a fuzzy controller it's proposed.

A *knowledge based system (KBS)* for close – loop control is a control system which enhances the performance, reliability, and robustness of control by incorporating knowledge which cannot be accommodated in the analytic model upon which the design of a control algorithm is based, and that is usually taken care of by manual modes of operation, or by other safety and ancillary logic mechanisms. In the context of this definition one can distinguish between two major classes of *KBS* for closed – loop control [2]:

*i)* one class where the *KBS* is involved in the supervision of the close algorithm;

*ii)* another class where the *KBS* directly achieves the closed loop operation, thus completely replacing the conventional control algorithm.

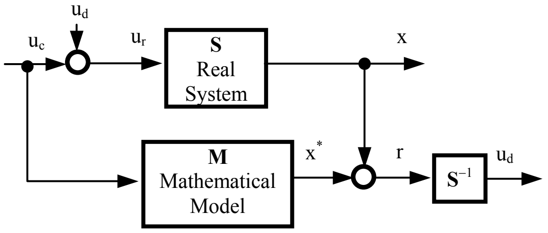


Fig. 2. Scheme for obtaining the error vector

To detect on - line all possible faults it was used a *KBS* for direct expert control (*DECS*), which means that *KBS* is used in a close loop, thus replacing completely the conventional control element.

To design such a *DECS* it was needed some implicit or explicit knowledge about the process to be controlled. Knowledge here means a model that provides a conceptual structure to capture those aspects of the process, which accurately represent its behavior.

For our diagnosis system this knowledge are those obtained from simulation the behavior of system without faults and with faults.

To design *DECS* it was used a well-known structure with the following blocks:

- *FM* – the *fuzzification module* which performs a *scale transformation* that maps the physical values of the current process state variables into a normalized universe of discourse and a so-called *fuzzification* which converts a point - wise (crisp), current value of a process state variable into a fuzzy set, in order to make it compatible with the fuzzy set representation of the process state variable in the previous rule.

- *Inference Engine* – there are two basic types of approaches employed in the design of a *DECS*: (1) composition based inference (firing) and (2) individual – rule based inference (firing).

- *Database* is to provide the necessary information for the proper functioning module, the rule base, and the defuzzication module.

- *DM – defuzzification module which has the functions* the so-called *defuzzification* that converts the set of modified control output values into a single point – wise value and the output *denormalization*, which maps the point – wise value of control output onto its physical domain.

## II. An Application Of Fault Execution Element Detection

The foregoing theory about a fault execution element based on the classical detection (analytic detection and localization theory) and on a modern one (fuzzy controller) is now illustrated by applying it to an helicopter with a dynamic in vertical plane characterized by the following matrix [5]:

$$\mathbf{A} = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.010 & 0.0024 & -4.0208 \\ 0.1002 & 0.3681 & -0.707 & 1.42 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

$$\mathbf{B} = \begin{bmatrix} -0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.52 & 4.49 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

where the vector state and command vector are composed by [3]:

$x_1 = u$ – the longitudinal velocity;
$x_2 = w$ – the vertical velocity;
$x_3 = \omega_y$– the rate of pitch (the pitch angular velocity);
$x_4 = \theta$– the pitch angle,

$u_1$ – the general cyclic command;
$u_2$ – the longitudinal cyclic command.

### *2.1 The Classical Approach of the Fault Detection and Localization for an Execution Element.*

It was considered that a fault could appeared at the execution level, that means at the servo-hydraulic motors which command the cyclic command and the longitudinal cyclic command (because of a low supply voltage, or mechanical brake or of a hydraulic leakage in a high pressure circuit or low pressure circuit, or mechanical brake). Practically the real command is altered with another constant command like in Fig. 3.

The helicopter has an unstable dynamic and for stabilizing it was applied the well-known quadratic controller, with the following penalization matrix:

$$\mathbf{Q}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\qquad \mathbf{R}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

With these the gain matrix is:

$$\mathbf{K}=\begin{bmatrix}0.9266&-0.0147&-0.9622&-1.3868\\-0.0225&-0.8448&0.1885&0.7135\end{bmatrix}.$$

also represented in Fig.3.

For this model (Fig.3) and using the fault execution element like a command, which affects the command input by vector:
$\boldsymbol{u}_{\mathbf{d}}=\begin{bmatrix}\mathrm{u}_{\mathrm{d}1} & \mathrm{u}_{\mathrm{d}2}\end{bmatrix}$
and with the inverse of $\mathbf{B}^{'}\mathbf{B}$ from (7) results:

$$u_{\boldsymbol{d}}(t)=\left(\boldsymbol{B}^{'}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^{'}\left[\dot{r}(t)-Ar(t)\right].$$

How in the equation above the residual vector $\boldsymbol{r}$(t) is known and for calculated his differentiate it was applied approximation with the difference of the first order, obtaining a recursive relation such as:

$$\boldsymbol{u}_{\mathbf{d}}(\mathrm{k}\,\boldsymbol{T})=\frac{1}{\boldsymbol{T}}\left(\mathbf{B}^{'}\mathbf{B}\right)^{-1}\mathbf{B}^{'}\left\{\left[\mathbf{I}-\boldsymbol{T}\mathbf{A}\right]\boldsymbol{r}(\mathrm{k}\,\boldsymbol{T})-\boldsymbol{r}\left[(\mathrm{k}-1)\boldsymbol{T}\right]\right\}$$

where $T$ is the sampling time.

Now it was assumed that the command vector $\mathbf{u}_{\mathbf{c}}$ is a 0.5 step, identically on the two components and something was happened on the first input, at the second 40 appeared a fault simulated by a step on the first input channel (that is practically the error vector $\mathbf{u}_{\mathbf{d}}$, which is unknown). In this case the simulation are represented in the following figures:

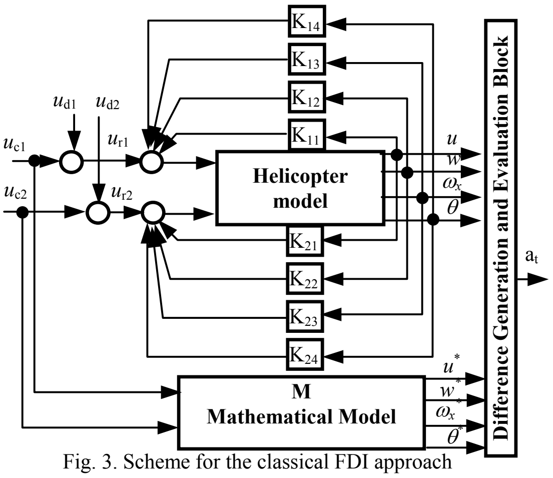
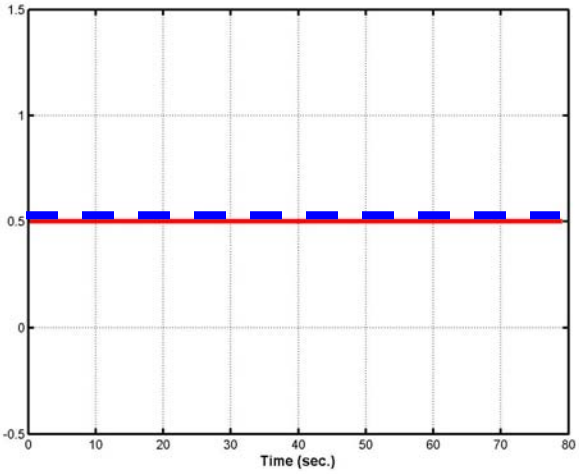


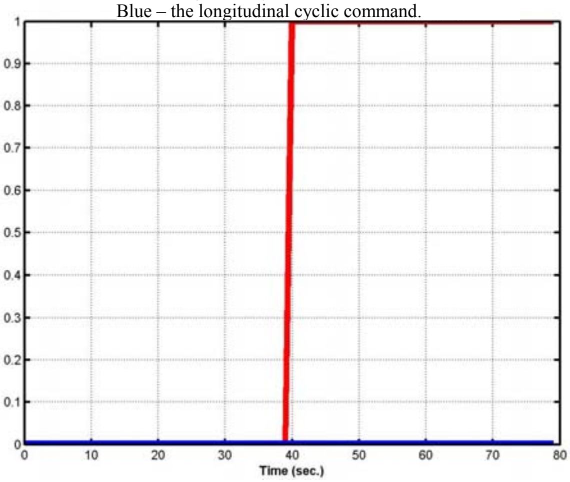
Fig. 3. Scheme for the classical FDI approach

- The command vector $\mathbf{u}_{\mathbf{c}}$:



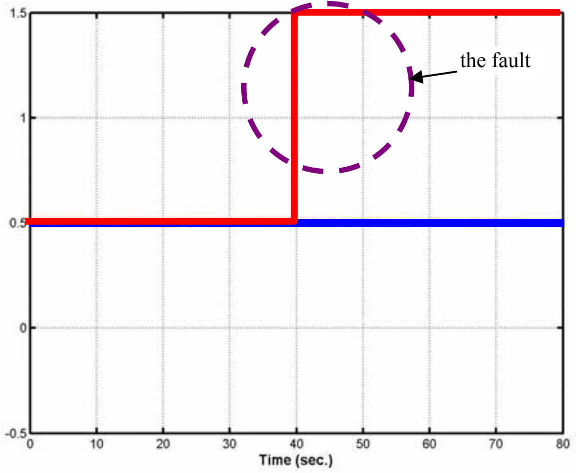
- The error vector which is unknown $\mathbf{u}_{\mathbf{d}}$ (the fault of execution element):

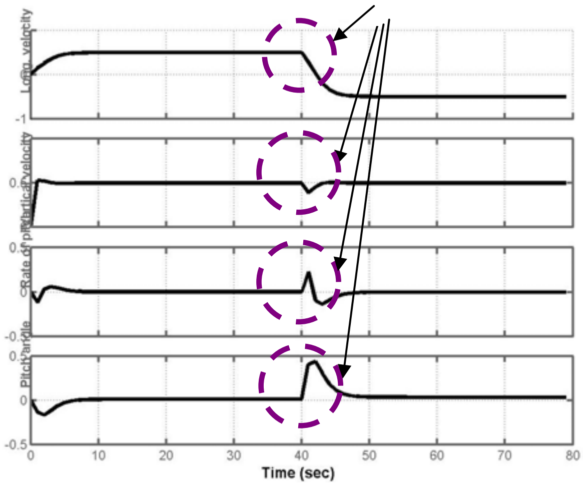
Red – the general cyclic command;
Blue – the longitudinal cyclic command.



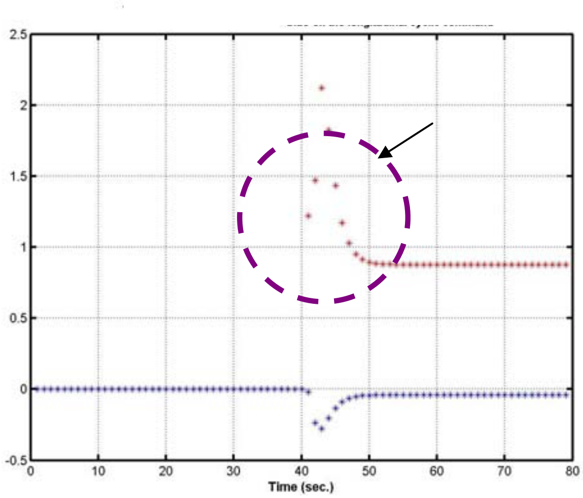
- The real command vector $\mathbf{u}_{\mathbf{r}}=\mathbf{u}_{\mathbf{c}}-\mathbf{u}_{\mathbf{d}}$ and the state vector:

Red – the general cyclic command;
Blue – the longitudinal cyclic command.





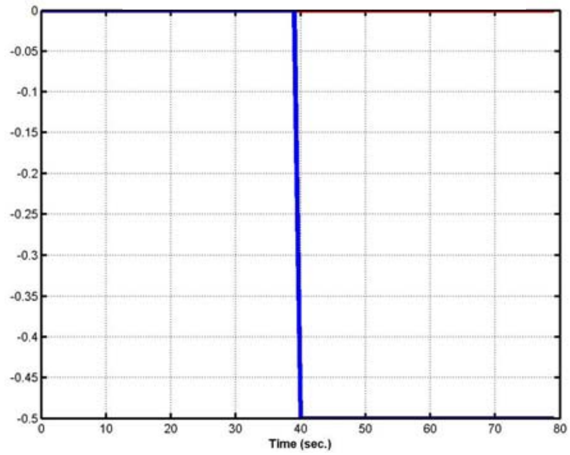
- The identification of the error vector:



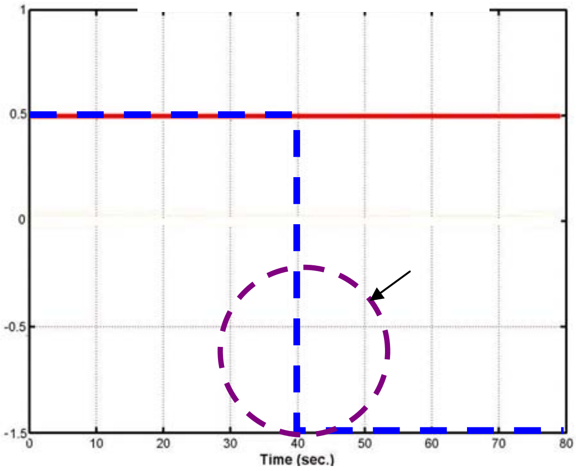
From the state diagram it can see that in the 40-th second when appeared the fault on the first command, that means the general cyclic command, the longitudinal velocity decrease and the other states remaining invariant. So the effect of the fault is like that another command. Also it can see that the two commands are practically uncoupled.

In the second simulation the vector $\mathbf{u_c}$ is the same like in the previous case and the fault is simulated by a step on the second input channel.

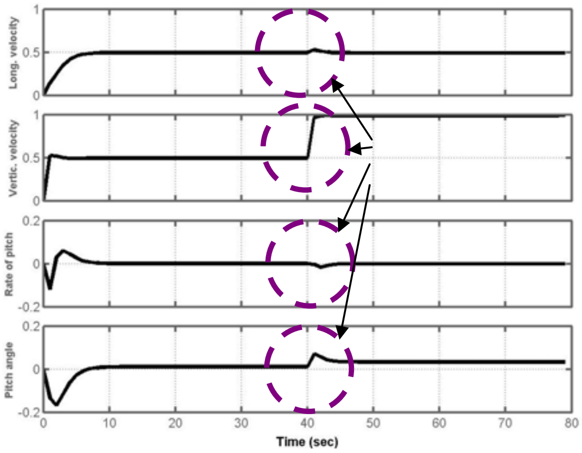
- The error vector which is unknown $\mathbf{u_d}$ which affect in this case only the second command (the fault of execution element):



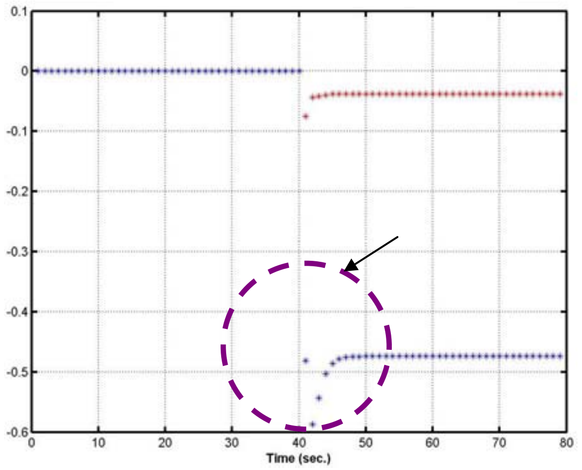
- The real command vector $\mathbf{u_r} = \mathbf{u_c} - \mathbf{u_d}$ and the state vector:



- The state vector:



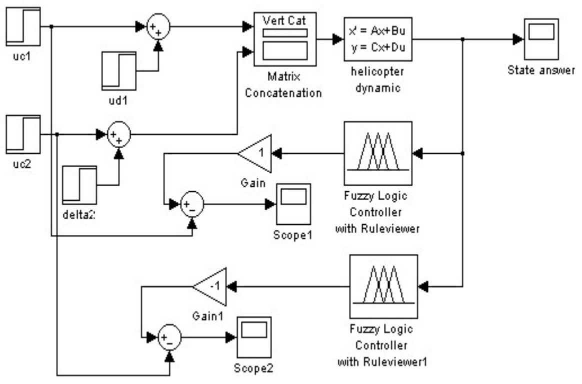
- The identification of the error vector:



After some simulation with the faults, which appeared on the execution elements of two command channels one can draw the conclusion that this classical method of identification works properly, only when the faults affected only one channel, otherwise the method is inconclusive.

2.2 *The Neuro -Fuzzy Fault Detection and Localization for an Execution Element.*

To solve this inconvenient in this paper fault identification using fuzzy controller is proposed. The structure of the process was represented in Section 1. A finally scheme that made the identification of the error vector $\boldsymbol{u}_{\mathbf{d}}$ which appeared like a fault, so it is unknown, is shown in the following figure.



To obtain the data for training a fuzzy logic controller it must be simulated the controller determined above (at the classical FDI approach) with faults (that means $u_{d1}\neq0$, $u_{d2}\neq0$) and without faults ($u_{d1}=0$, $u_{d2}=0$). The commands of this scheme (as seen in Fig.3) are $u_{c1}$ and $u_{c2}$.

The disturbed commands ($u_{d1}$, $u_{d2}$) that are in fact the faults could alter the commands ($u_{c1}$, $u_{c2}$) only in three ways: in a few seconds after a command was given the total command ($u_{c1} - u_{d1}$, or $u_{c2} - u_{d2}$) might be:

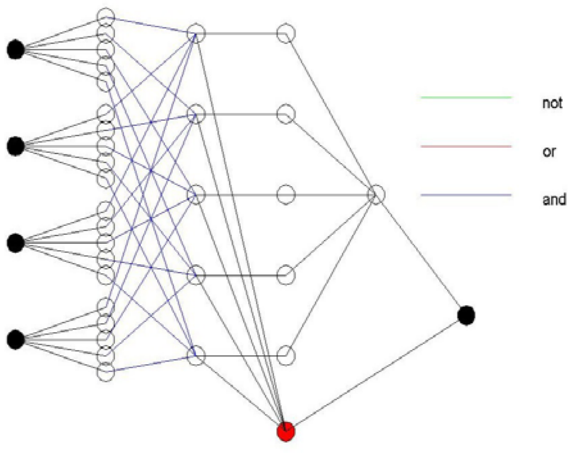
*i)* 0 if the execution element is broken;

*i)* between 0 and $u_{c1,\max}$ for the general cyclic command or between 0 and $u_{c2\max}$ for the longitudinal cyclic command (for example due to the low supply voltage at the servo-hydraulic motors);

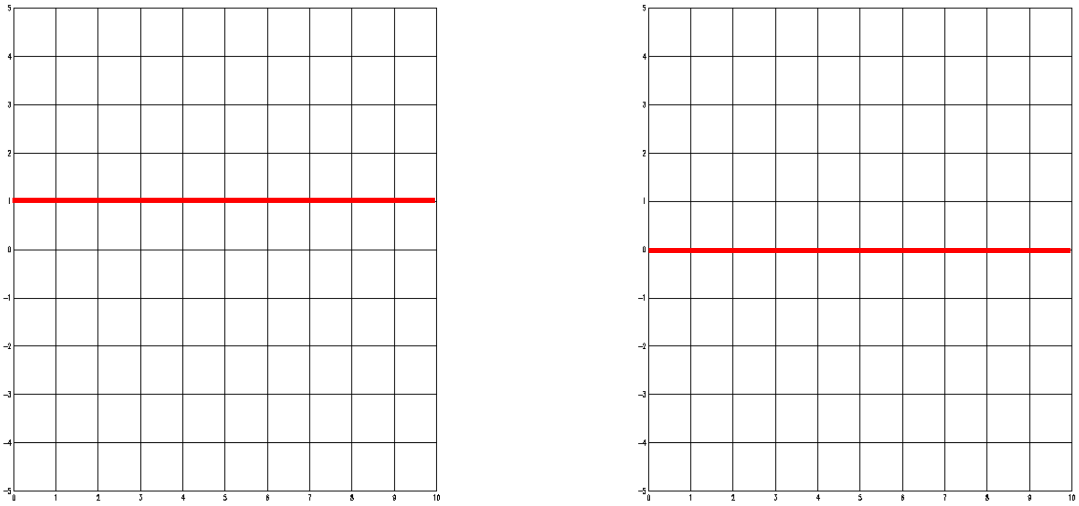
*ii)* between 0 and $-u_{c1,\max}$ for the general cyclic command or between 0 and $-u_{c2,\max}$ for the longitudinal cyclic command (for example due to the inverse supply voltage, in case of a converter broken).

With these data the fuzzy logic controller was trained with the aide of Matlab Anfis Program in 40 epochs. Practically the both two Fuzzy Logic Controllers are the same structure like:

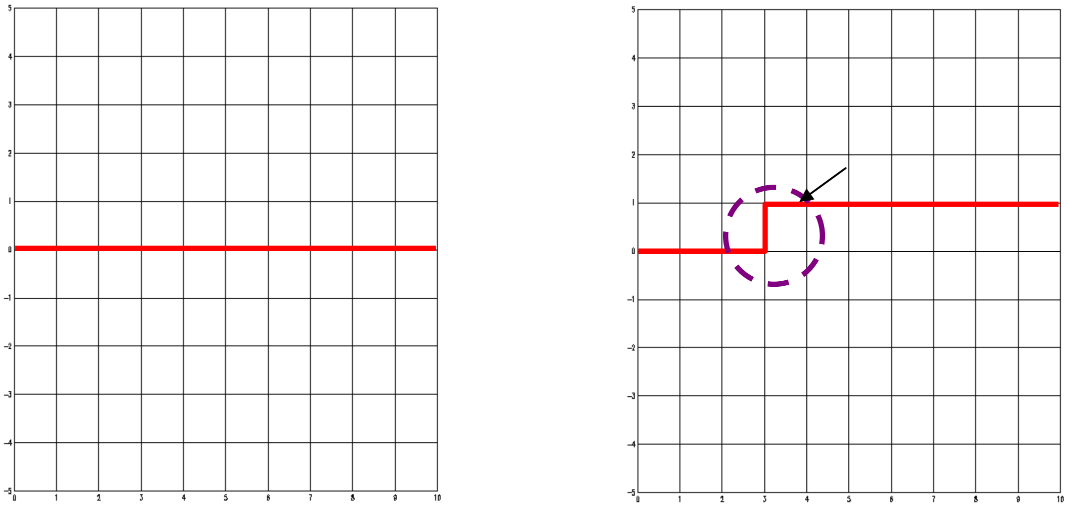
In the following figures is represented a simulation with the Simulink Model above.



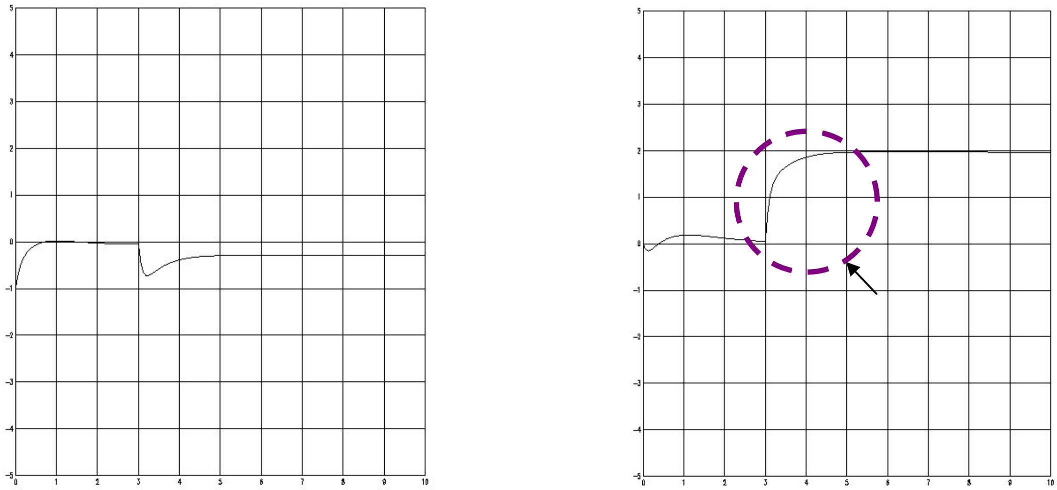
- The command vector $\mathbf{u}_c$:



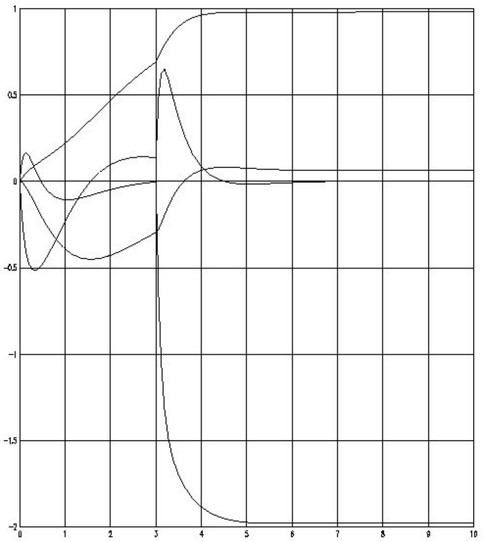
- The error vector $\mathbf{u}_d$ which is unknown (the fault of execution element):



- The identification of the error vector:



- The state vector:



## III. CONCLUSIONS

The difficulty in the state identification of both methods is due to the connection between the two commands, the general cyclic command and the longitudinal cyclic command. But because of the small longitudinal velocity of helicopter (that is between 100 – 150 m/s) the results of this connection is not so critically. Practically from the results of the simulation at this velocity it can say that the two commands are uncoupled, so the general cyclic command affects only the first state – the longitudinal velocity and the longitudinal command affects only the second state – the vertical velocity.

In conclusion comparing the results obtained with the two methods, a classical FDI of the execution element and a modern FDI with aide of the Fuzzy Controller, the last one is more adequate to identify the faults that appeared on the input command.

## REFERENCES

[1] J. Akermann, *Robust Control. Systems with Uncertain Physical Parameters*, Springer Verlag, pp.36-41, 1993.

[2] D. Driankov, H. Hellendoorn and M. Reinfrank. *An Introduction to Fuzzy Control*, Springer Verlag, Berlin, 1993.

[3] P.C. Hughes, *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, 1986.

[4] R.J. Patton and J. Chen, "Observer-based fault detection and isolation: robustness and application", *Control Eng. Practice*, vol.5, pp. 671-682, 1997.

[5] S.N. Singh and A.R. Coelho, "Nonlinear Control of Mismatched Uncertain Linear Systems and Application to Control Aircraft", *Journal of Dynamic. Syst. Meas. Contr.,* vol. 106, pp. 203-210, 1984.

[6] M. Staroswiecki, V. Cocquempot and J. Cassar, "Observer based and parity space approaches for failure detection and identification", *Proc. of IMACS symposium on Modeling and Control of Technological Systems,*, vol. 2, pp. 536-541, 1991.