King Fahd Univesity of Petroleum & Minerals

System engineering department

EE 550 Linear Control Systems

Dr. Samir A. Al Baiyat

Fault tolerant control

TERM PROJECT
FINAL REPORT

BY

Elrayyah Ali Yaseen   ID 260424

May 2007

Fault tolerant control

***Fault tolerant control** is one of the most active research areas in control theory, because a very important property in any control system is to perform it task reliably, and every system is vulnerable to faults, then if the reliability of the system can be preserved even under the faulty cases then more trust can be put on these systems. The advances in the branches of control theory like robust control, adaptive control, optimum control, intelligent control, fault detection and diagnosis, control reconfiguration and parameter estimation make it possible to do a better job in the fault tolerant control (FTC) field. Each of these branches contributes a lot to solve this problem the thing that gives an opportunity for every one in these fields to contribute from his side. For that reason this becomes very active area*

In this report a tried is made to introduce the major parts of this topic, and to give an overview of some of the work done in each area then a modification for some previous work is introduced, and verification of these modifications made through a simulation

Its very important to concentrate on the point that the building a Fault tolerant control system (FTCS) must take place in all stages s of the system design and should be treated by looking at the overall picture, then to combine the analysis of fault diagnosis and detection (FDD) part side by side with the control reconfiguration parts. The supervision and the previous analysis of all possible faults will make the problem as a constrained optimization problem to be solved.

Key words: fault tolerant control, Fault diagnosis, control reconfiguration, state estimator, parameter estimator

1-Introduction

1.1-What is fault tolerant control

A fault-tolerant control system (FTCS) is a control system that possesses the ability to accommodate for system faults automatically and to maintain overall system stability and acceptable performance in the event of component failures [1]

Typically, a FTCS consists of three parts: a reconfigurable controller, fault detection and diagnosis (FDD) scheme, and a control law reconfiguration mechanism. Key challenges are to design: *(a)* a sufficiently robust controller which is reconfigurable, *(b)* a FDD scheme with high sensitivity to faults and low sensitivity to disturbances, and *(c)* a reconfiguration mechanism which can recover the pre-fault system performance as much as possible. [1]

The remaining parts of this report are organized as follows, section 2 introduce an important concept related to FTC, section 3 cover FDD related issues and those related to control reconfiguration in section 4, section 5 shows a modification on a previous work for FDD and combine it with a suitable control reconfiguration and the result of the made simulation were analyzed, finally a conclusion is made for the report

2- Important concepts related to FTC

2.1- the concept of redundancy

The redundancy can be introduced by a relation that can be used to produce the same effect of some part of the system by using the remainder system components. Redundancy analysis is the process to see what are the available redundancy in the system and their degree and how can we use then for fault diagnosis and accommodation, the redundancy analysis and design is one of the first stages in designing fault tolerant system.

There two types of redundancy namely physical and analytical redundancy.

2.1.1- physical redundancy

The traditional engineering approach to achieve fault tolerance control in dynamic systems is through the use of hardware redundancy e.g. redundancy for sensors can be done by putting 2,3,4..,n sensors and by taking the majority voting. For actuators to have standby actuators that will replace the faulty one

This approach constructs redundant physical subsystems. However, the additional cost, space and complexity of incorporating redundant hardware makes this approach unattractive. [2]

2.2.2- analytical redundancy

By using state estimator or parameter estimator we can generate a value that is very close the real fault-free value. Here instead of using physical redundant component analytical redundancy can be used, it is based on using the model of the system (state space, ARMAX, transfer function) to try to predict the output (or the state) according to this input this prediction can be treated as a redundancy in the system. In aeronautical applications, there has been an increasing tendency to not substitute physical redundancy entirely by the analytical alternative, but to suppress some index of redundancy e.g. in sensor faults the analytically-derived signal can be used instead of the impaired sensor signal, perhaps under limited authority. [3]

The success of the analytical redundancy approach is heavily dependent on the quality of models [2].

But in some situation the use of the analytical redundancy is not sufficient to get the required result so the physical redundancy should be used, and also a combination of them may also be applicable

2.2.3- how to measure the redundancy

It will be very helpful if we can find a way to measure the available redundancy in the system, then in cases of faults this information can be used to know what other components that can be modified to regain the pre-fault situation.

In [4] the author uses the eigenvalues of the product of the controllability and observability Gramiuans to check the ability to configure the system in case of faults by checking the least positive value σ_{min} and the effect in the change in each system parameter in this value to detect what he calls reconfigurability which indicate the ability to reconfigure the remaining system to regain the normal operation.

2.2- sensor placement

In some cases after making the redundancy analysis it may be found that if an additional sensor in a specific place is added it will give very helpful information then we have to select the best place a sensor to detect the as more fault as we can e.g. in closed loop according to [10] by using the concept of Bode sensitivity the authors found that by taking the command from the controller more valuable information can be got than those from the output signal.

3- Fault Detection and diagnosis "FDD"

3.1- Concepts of FDD

The fault detection and Diagnosis FDD is a part of the system that is responsible of the detection of the occurrence of any fault then to determine in which component the fault is and finally the magnitude of the fault, its components are shown in figure 1

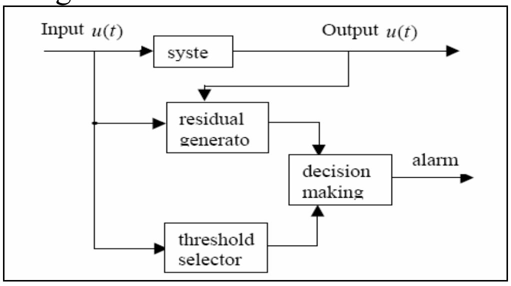

Figure1 FDD system

According to these points we can divide the FDD task into three parts: fault detection, isolation and estimation

3.1.1- fault detection

There are many ways to detect the fault, but the most common way is to use analytical redundancy to produce the expected output (or state) then to compare this value with the actual one, if the difference "called residual" exceeds a predefined threshold then a fault will be declared. This threshold depends on the noise level in the system e.g. if the noise is Gaussian noise with zero and variance σ then to get threshold with probability of error 10^{-4} then we have to look at the probability distribution of this Gaussian noise and what is the value above which the noise has the probability of 10^{-4} this value will be our threshold. The way to put threshold may form a weak point in the detection of a fault according to [6], also the threshold value could vary with time according to value of input signal as shown in figure 2

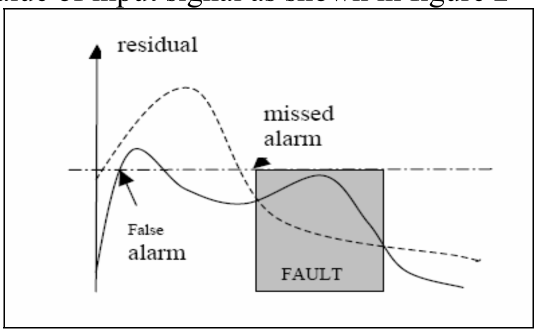

Figure 2 variable threshold

3.1.2- fault isolation

The objective here is to isolate faults. This is achieved by generating structured residuals sensitive to certain faults and insensitive to others [5]. Here we can use state, parameter or parity estimators to isolate the faults:

3.1.3- fault estimation

After the fault is isolated in a set of component (or in a single component) the next step is to estimate the value of this fault by checking the values or the statistical properties of the generated residuals

3.2-robustness of FDI

When designing FDI it should be sensitive to faults but not for disturbances noise model mismatch, this can be obtained by e.g. designing a good Kalman gain, using good identification methods

The fault estimation problem use the ideas of the robust control optimization and fault estimation designs are best combined for example using $H\infty$ optimization. [3]

3.3- types of FDI

3.3.1- Model based

Here the FDI will use a model for the system to generate a prediction for the signals or to estimate the parameter to be compared with the real values and detect any fault, then the model of the system dynamics will give additional source of information (over the inputs and outputs) which will help for fault diagnosis, of course the model must be accurate to give good results, most of the FDI systems are based on this type, generally state and parameters estimator are commonly used:

3.3.1.1- state estimators

The basic idea here is to build an observer that contain a mathematical model of the system, by taking the input signal an then calculate the state values according to the model [13] the model here is either discrete time system or disceretized continuous system to use the algorithm in computer for estimation.(Figure 3)

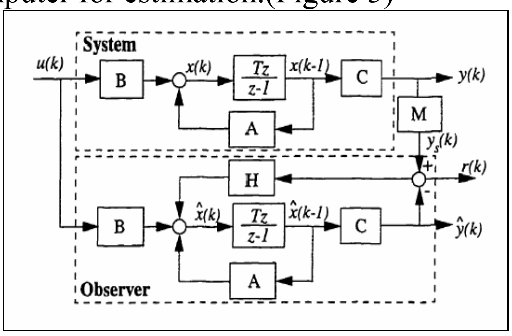

Figure 3 state estimator

Here the estimated state $A\hat{x}[K]$ "using Kalman filter" is given by:

$$\hat{x}[k+1\backslash k] = A\hat{x}[K/K] + Bu[k] + w[k]$$

$$\hat{x}[k+1\backslash k+1] = \hat{x}[k+1\backslash k] + g(y[k+1] - C\hat{x}[k+1\backslash k])$$

$$e[k+1] = x[k+1] - \hat{x}[k+1\backslash k+1]$$
$$= A[k] + Bu[k] + w[k] - A\hat{x}[k\backslash k] + Bu[k] + g(Cx[k+1] + v[k] - C\hat{x}[k+1\backslash k])$$

$$g(Cx[k+1] + v[k] - C\hat{x}[k+1\backslash k]) = g(C(Ax[k] + Bu[k] + w[k]) + v[k] - C(A\hat{x}[k+1\backslash k] + Bu[k]))$$

$$e[k+1] = (I - gC)A(x[k] - \hat{x}[k\backslash k]) + (I - gC)w[k] + gv[k]$$

$$e[k+1] = (I - gC)A(e[k]) + (I - gC)w[k] + gv[k]$$

$e[k]$ is the state estimation error $w[k]$ is the disturbance $v[k]$ is the measurement noise, squaring both side and taking the mean

$$\sum_{ee} = (I - gC)A\sum_{ee} A^T(I - gC)^T + (I - gC)\sum_{ww}(I - gC)^T + g\sum_{vv} g^T \text{....(1)}$$

$\sum_{ee}$ the covariance of the state estimation error

$\sum_{ww}$ the covariance of the plant disturbances

$\sum_{vv}$ the covariance of the measurements noise

Then by taking the first order variance we find

$$(A\sum_{ee} A^T + \sum_{ww})C^T = g[C(A\sum_{ee} A^T + \sum_{ww})C + \sum_{vv}] \text{..(2)}$$

By solving (1) and (2) we can find $\sum_{ee}$ and g. Then by calculating $e[k]$ we can detect the fault, when there is no fault $e[k]$ will be close to zero and its value will be due to noise, but in the case of fault the value of $e[k]$ will be considerably high because the parameter that produced $x[k]$ is different from those which produced $\hat{x}[k]$

3.3.1.2- parameter estimators

Another way for fault detection and residual generation is to use identification methods to make estimation for the system parameters and compare there estimate with nominal parameters if there is some fault then the estimate will be different from the nominal one and a residual will be generated figure 4 e.g. in [7]

$$z[k] + a_1 z[k-1] + a_2 z[k-2] + ... + a_n z[k-n] = b_1 u[k-1] + b_2 u[k-2] + ... + b_n u[k-n]$$

$$y[k] = z[k] + v[k]$$

$$y[k] = \psi^T[k]\theta^* + w[k]$$

$y[k]$ is the output $u[k]$ is the input θ^* is the system parameters. Then the RLS technique is used to make parameter estimation on line

$$\hat{\theta}[k] = \hat{\theta}[k-1] + P[k]\psi[k](y[k] - \psi^T[k]\hat{\theta}[k-1])$$

$$P[k] = \left(P[K-1] - \frac{P[k-1]\psi[k]\psi^T[k]P[k-1]^T}{1+\psi^T[k]P[k-1]\psi[k]} \right)$$

$p[k] \in R^{2nx2n}$ *is a positive definite symmetric matrix that be also written as* :

$$P^{-1}[k] = \sum_{i=1}^{k} \psi[i]\psi^T[i] = P^{-1}[k-1] + \psi[k]\psi^T[k]$$

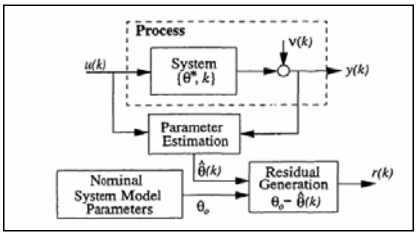

Figure 4 parameter estimator

The key issue of this technique is to ensure the convergence of the result, in [7] the author discussed this issue and he show that it will convergence rate is 1/k.

In the presence of noise the estimator may be biased (due to inaccurate identification method) so to avoid false fault detection this bias must be compensated by e.g. adjust the threshold or other similar techniques here the residual will represent fault detection isolation and estimation. This approach may take long time to get consistent estimation for the parameter specially if the number of parameter is large and its required an input signal with specific degree of persistence excitation the thing that may not be available with the input signal, one way to overcome this is by superimpose perturbation signals with the input that is persistently exciting enough, However, in many practical applications, it is will not be acceptable to apply additional perturbation signals.

Of course a combination of state and parameter estimator could be used to get the advantages of each one. To solve these problems there is a new approach which will give faster result which is more accurate its called variable length sliding window block-wise least squares VLSWBLS[11], here instead of using RLS a every time a block of input output set "data from the time k-L up to current time k" will be used to calculate the least squares LS estimate of the parameter, and when a new set of data becomes available it will replace one from the old block, in this way it has better noise suppression ability than RLS, then if a fault is detected the block size will shrink to zero to discard all the old data "which does not have any information about the faulty parameter" and the block will start to expand containing only the post-fault information which will give better estimation. This procedure will give faster convergence, and after some modification (by using instrumental variables method) can be used for white and colored noised this could be very helpful if combined with state estimator method

4- Control reconfiguration for FTC

In FTC Control reconfiguration process is to redesign the controller by changing the gains for the actuators, to use estimators for the output or to combine input in such a way that will compensate (at least partially) the fault effect, by looking to the eigenvalues the stability or the validity of the operating points or the performance availability safety reliability figure 5

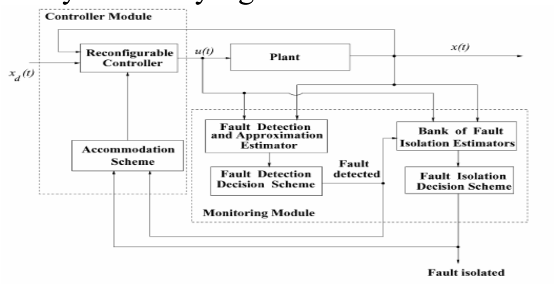

Figure 5 FTC system

4.1- Concepts

In [8] and the [14] there are shown different ways to analyze the effect of fault in system that have multi subsystems interconnected and forming a larger system, then the FTC should deal with the effect of faults in each subsystems and then the effect of this fault in the other subsystems, they call it fault propagation analysis, also its very important to analyze the redundancy in the system e.g. in [15] they use graphical method obtained from the system equation to find the possible redundancy in the system, in [4] and also the method mentioned earlier by checking the eigenvalue of the product of the controllability and observability Gramians

4.2- control reconfiguration

Before discussing control reconfiguration its important to show that FTC systems can be divided into active and passive FTC, in passive FTC there is no FDI or control reconfiguration, not like the case in active FTC which is build on thses two components. Here an overview of passive and active FTC is given:

4.2.1- passive FTC

Here the robust control techniques are used to make the system robust against some faults [7]. A passive fault tolerant control system is a control system designed to tolerate system component failures with acceptable performance degradation, without changing the control strategy or parameters. One of the advantages of properly designed passive FTCS is that the robustness against component failures has already been built into the system. In other words, the fault scenarios have already been taken into account at the controller design stage. However, the design procedure for such controllers is often more involved because the control system has to perform satisfactorily not only during normal system operation but also under various component failures.

In general, passive fault-tolerant control does not involve the joint estimation of control and fault signals. However, in many practical situations, the use of robust control alone to achieve fault tolerance may be quite a risky thing to do. As a non-intelligent controller, without the use of diagnostic information and with no knowledge of fault occurrence - where and how serious the fault is – the passive system will have a very limited fault-tolerance capability. Basically, the passive controller will reject the fault only if can be de-sensitized to the fault effect just as if it were a source of modeling uncertainty [3].

4.2.2- active FTC

In contrast to the passive FTC here the fault tolerance is not build in the controller but it depends on the following steps

- To detect isolate and estimate the fault value
- To use these information to modify the system (in most cases the controller) in a way that will give acceptable result

The main ideas here are

-how to build robust FDD system

-how to reconfigure your design in a way that will preserve the stability of the system

-how to combine control reconfiguration and FDD

-how to do all that in a short and acceptable time period [7].

In most cases the researchers are concentrating in dealing with first two points separately and the efforts now is to try to deal with the combination of both of them Here are some examples of the most used techniques for the control reconfiguration which can be used for active FTC

4.2.2.1- pseudo-inverse method (PIM)

The goal of Pseudo Inverse Method is to find controller feedback gain which would minimize the difference between the closed-loop dynamics of the nominal and faulty systems.

To explain the idea of this method let a system has state feedback like

$u(t) = Kx(t)$

Then when a fault occurred the system will be

$\dot{x}(t) = A_f x(t) + B_f u(t)$

Therefore, in order to recover the nominal system performance, it is highly desirable that the reconfigured closed-loop system matrix after reconfiguration be the same as that of the nominal system, i.e.

$A_f + B_f K_r = A + BK$

In this method the approximate solution is

$K_r = B_f^+ (A - A_f + BK)$

This selection will minimize the following criteria

$J = \left\| (A_f + B_f K_r) - (A + BK) \right\|$

One of the advantages of this method is its simplicity in calculating the reconfigurable control law, however this method could not ensure stability, some research tried to improved the method by dealing with it as a constrained optimization problem, constraints must be put so as to stabilize the eigenvalues of the failed system [9], but in many situation this constrains will be very restrictive for practical use [7], also there is an assumption in this method that perfect information on post-fault system is available. In other words, such a method is only for the controller redesign based on known failure information.

4.2.2.2- model following

In this method we want the fault system states to follow the nominal system then if the nominal system is

$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t)$

Then the structure of the reconfigured system is as shown in figure 6

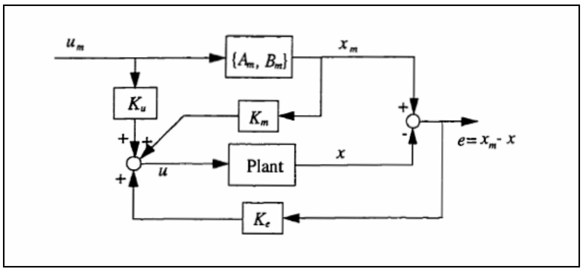

Figure 6 model following

The value of $e(t)$ which is the error in the states is

$$\dot{e}(t) = (A_f - B_f K_e)e(t) + (A_m - A_f)x_m(t) + B_m u_m(t) - B_f[K_m x_m + K_u u_m]$$

K_e is a stabilizer gain for system then to minimize the following criterion

$$\min J_m = \min\left\|\dot{e} - (A_f - B_f K_e)e\right\|_2 = \min\left\|(A_m - A_f)x_m(t) + B_m u_m(t) - B_f[K_m x_m + K_u u_m]\right\|_2$$

The following values can be used

$$K_m = B_f^+(A_m - A_f)$$

$$K_u = B_f^+ B_m$$

$$\dot{e}(t) = (A_f - B_f K_e)e(t)$$

Then K_e should be selected to make the eigenvalue of the above equation stable then as $e(t) \to 0 \; or \; x(t) \to x_m(t) \; as \; t \to \infty$

By this way the difference between the normal state x_m and faulty state will be bounded and the system will be BIBO stable as long as the normal system is stable

Model following can grantee the complete recovery of the system stability and partially recovery of the performance [7].

4.2.2.3- Eigen-structure assignment

The main idea here is to recover the system eigenvalues and eigenvectors. This method not only guarantees the stability of the reconfigured system by recovering the system eigenvalues, but also recovers the system performance to the maximum extent by placing the eigenvectors as close to those of the original system as possible.

Then suppose the nominal closed loop system is

$$\dot{x}(t) = (A + BK_0)x(t)$$

And the faulty system is

$\dot{x}(t) = (A_f + B_f K_f)x(t)$ Here K_f is to selected to place the eigenvalue of the faulty system as close as possible to the nominal one $\lambda\{A_f + B_f K_f\} = \{\lambda_i, i = 1,2,...,n\} = \lambda\{A + BK_0\}$ Also the K_f should make the eigenvector of the faulty system as close as possible to those of the nominal one then

$$(A_f + B_f K_f)v_i^f = \lambda_i v_i^f$$

Then

$v_i^f = (\lambda_i I - A_f)^{-1} B_f K_f v_i^f$ The value of v_i^f that will minimize the following criteria

$Min(v_i - v_i^f)^T(v_i - v_i^f)$ is

$$v_i^f = S_i\left(S_i^T S_i\right)^{-1} S_i^T v_i$$

Where

$$S_i = (\lambda_i I - A_f)^{-1} B_f$$

Then K_f should be selected to get this v_i^f

It is clear that the main advantage of this method over the PIM and the mode1 following method is that it can guarantee the complete recovery of the system stability and partially recovery of the system performance.

However, this method also requires that the post-fault system information is known or provided by a perfect FDI scheme [7]. Also The effects of non-matching of eigenvectors to that of in nominal condition were not explained [9]

4.2.2.4- artificial intelligence

Since active FTCS involve real-time FDI and controller reconfiguration, it appears to possess certain degree of intelligence. Therefore, such types of systems can use intelligent control systems [7]. Here by using artificial intelligence and fuzzy logic to keep some previous knowledge about the fault, and to use neural network for the recognition of the fault, FTC can be build, e.g. when we deal with nonlinear and\or system without valid model the use of these technique will be very helpful.

5- Simulation

5.1- The system

The discussion here will be about the use extended Kalman filter [12] and [13] with some modification to detect and isolate possible faults in the actuators then to use identification techniques to identify the value of the fault

5.1.1- The model

The system has the following state space representation " taken from [16] with some modification"

$x[k+1] = Ax[k] + Bu[k] + w[k]$

$y[k] = Cx[k] + v[k]$

$x[k]$ is the state variable vector nx1 here n=3, A is the state matrix nxn, B is the actuator matrix nxp here p=2, $u[k]$ is the input vector px1, $w[k]$ is system disturbances white noise with zero mean and variance λ_w, $y[k]$ is measured output qx1 q=2, C is the sensor matrix qxn, $v[k]$ is the measurements noise white noise mean = zero with variance λ_v

$$A = \begin{bmatrix} .9 & 1 & 0 \\ 0 & .95 & 0 \\ 0 & 0 & .95 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

In this example λ_w=0.1 and λ_v =0.01

The input is used as unit step u_1=2 u2=$_1$

5.1.2- possible faults

Here the concentration is more on the actuator faults than sensors faults, for sensors case similar approach can be done. The actuator fault can be represented by a factor γ_{aj} which is multiplied by the actuator corresponding column e.g. if a fault happened in the second actuator then the actuator matrix will be

$$\begin{bmatrix} b_{11} & \gamma_{a2}\, b_{12} \\ b_{21} & \gamma_{a2}\, b_{22} \\ b_{31} & \gamma_{a2}\, b_{32} \end{bmatrix}$$

If γ=1 then there is no fault if γ=0 there is a complete failure if 0<γ<1 then there is a partial fail if γ=K/u_2 this is a way to simulate a state of blocked actuator in a state that doesn't respond to the actuator input

Similarly we can represent the faults in the sensors by γ_{si} then a fault in the first sensors will be represented by

$$\begin{bmatrix} \gamma_{c1}C_{11} & \gamma_{c1}C_{12} & \gamma_{c1}C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix}$$

5.2- Kalman filter and the extended Kalman filter

In this example state estimator is used for fault detection and isolation because according to many paper e.g. [7] the state estimator always gives fast fault detection and by using the concept of extended Kalman filter in [12] and [13] fast fault isolation can be got, their idea is used here with some modification, then because state estimation suffer from slow fault identification, we can get benefit of the parameters estimation techniques to identify the fault faster.

5.2.2- extended Kalman filtering

In [13] and [12] they used the Kalman filter to detect isolate and identify the faults, here the same concept is used but with some modification

According to [12] and [13] the matrix g "equation (1) and (2) " is to be selected in a way that it will be insensitive to specific fault group, here g is to be selected to isolate only one fault then for each expected fault its required to make a Kalman filter.

Generally to detect and isolate a fault that could take place in the ith actuator (which is represented by the parameters in the ith column in the B matrix) two hypothetical matrices have to be formed

$$B_{inci} = \begin{bmatrix} b_{11} & \cdots & b_{1\,i-1} & 0 & b_{1\,i+1} & \cdots & b_{1p} \\ b_{21} & \cdots & b_{2\,i-1} & 0 & b_{2\,i+1} & \cdots & b_{2p} \\ b_{31} & \cdots & b_{3\,i-1} & 0 & b_{3\,i+1} & \cdots & b_{3p} \\ . & \cdots & . & . & . & \cdots & . \\ b_{n1} & \cdots & b_{n\,i-1} & 0 & b_{n\,i+1} & \cdots & b_{n\,p} \end{bmatrix}$$

$$B_{exci} = \begin{bmatrix} 0 & \cdots & 0 & b_{1\,i} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_{2\,i} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_{3\,i} & 0 & \cdots & 0 \\ . & \cdots & . & . & . & \cdots & . \\ 0 & \cdots & 0 & b_{n\,i} & 0 & \cdots & 0 \end{bmatrix}$$

It's very clear that B=B_{inci}+B_{exci} and if there is a fault in the ith actuator so that the b_i column become γ_{ai} b_i then B_f (which is the value of B after the fault) can be written as B_f=B_{inci}+ γ_{ai} B_{exci}

Then the selection of g_i to detect and isolate a fault in b_i should satisfy "as what will be explained latter": $(I - g_i C)B_{exci} = 0 \quad or \quad g_i C B_{exci} = B_{exci}$

To get g_i that satisfy this equation according to [13] and [12] then C should has a rank at least equals to the rank of B_{exci} but here the rank of B_{exci} is 1 because all the columns are zero except one then any nonzero C will satisfy the equation

The equation

$(I - g_i C)B_{exci} = 0$

by putting $K = (I - g_i C)$ *then* $K\ B_{exci} = 0$

Here K in n x n matrix then here is infinite number K's that satisfy the equation and this imply infinite number of g_i matrices, one should notice that we have to select g_i that will make the eigenvalues of the matrix $(I - g_i C)A$ as small as possible to make the error in the state estimation "as what will be shown latter" decay very fast to reduce the effect of the error between the estimated and real state values also the value of g_i should reduce the effect of w and v in the estimation error. In [12] the author used the error between the real and the estimated parameter to detect and identify the fault, but here we are not going to use e[k] to detect, isolate and estimate the fault then the requirement in g_i is just to detect and isolate not like [12] who use e[k] to estimate the fault then any acceptable g_i can be used

5.2.3-1 the detection and isolation

In many papers [12] [7] they uses e[k] to make both the fault detection isolation and estimation but $x[k]$ can not always measured then to get $e[k]$ is possible in simulation but in real systems we can use only the values of $y\ u\ and\ \hat{x}$ to make the fault detection and isolation here these values will be used to detect and isolate the faults. An illustrative example is given here; to detect a fault in the i[th] actuator first estimation for $x[k]$ is made by using B_{inci} only and ignoring the noise (for simplicity) then

$$\hat{x}[k+1\backslash k] = A\,\hat{x}[k/k] + B_{inci}\,u[k]$$

$$here\ \ by\ \ assumeing\quad \hat{x}[k/k] = x[k]$$

$$(this\ \ assumption\quad will\ \ be\ \ proved\quad after\quad some\quad steps\ )$$

$$then\quad \hat{x}[k+1/k] \approx x[k+1] - B_{exci}\,u[k]$$

$$this\ \approx is\ \ due\ \ to\ \ the\ \ noise$$

$$\begin{aligned}\hat{x}[k+1/k+1] &= \hat{x}[k+1\backslash k] + g_i(y[k+1] - C\hat{x}[k+1\backslash k])\\ &= \hat{x}[k+1\backslash k] + g_i(C(Ax[k] + Bu[k])\\ &\quad - C(A\hat{x}[k/k] + B_{inci}\,u[k]))\\ &= \hat{x}[k+1\backslash k] + g_i(CA(x[k] - A\hat{x}[k/k)\\ &\quad + C(B - B_{inci})u[k])\\ &= \hat{x}[k+1\backslash k] + g_i(CA(e[k]) + CB_{exci}\,u[k])\end{aligned}$$

$$\begin{aligned}\hat{x}[k+1/k+1] &= \hat{x}[k+1\backslash k] + g_iCB_{exci}\,u[k] + g_i(CA(e[k])\\ &= \hat{x}[k+1\backslash k] + B_{exci}\,u[k] + g_i(CA(e[k]) \approx x[k+1]\end{aligned}$$

by suitable selection of g_i any difference due to the initial condition will decay and after some time $\hat{x}[k/k] \approx x[k]$ then by checking the value of $y[k+1] - C\hat{x}[k+1/k+1]$ this value will be random value with zero mean because of *w[k]* and *v[k]* but with using acceptable g_i and $(I - g_iC)$ this value will be very small and by selecting a suitable threshold this value can be used for fault detection and isolation. The idea here is if there is any fault in the i[th] actuator the value $y[k+1] - C\hat{x}[k+1/k+1]$ will not be affected to see this suppose there is a fault in the i[th] actuator then:

$$x\,[k+1] = Ax[k] + B_{inci}u[k] + \gamma_{ai}B_{exci}u[k] + w[k]$$

$$y[k] = Cx[k] + v[k]$$

$$\hat{x}[k+1\backslash k] = A\,\hat{x}[k\backslash k] + B_{inci}u[k]$$

$$\begin{aligned}\hat{x}[k+1/k+1] &= \hat{x}[k+1\backslash k] + g_i(y_f[k+1] - C\hat{x}[k+1\backslash k])\\ &= \hat{x}[k+1\backslash k] + g_i(C(Ax[k] + B_{inci}u[k] +\\ &\quad \gamma_{ai}B_{exci}u[k] + w[k]) + v[k] - C(A\hat{x}[k/k] + B_{inci}u[k]))\\ &= \hat{x}[k+1\backslash k] + g_i(CA(x[k] - A\hat{x}[k/k) +\\ &\quad C(B_{inci} + \gamma_{ai}B_{exci} - B_{inci})u[k] + Cw[k] + v[k])\\ &= \hat{x}[k+1\backslash k] + g_i(CA(x[k] - A\hat{x}[k/k) +\\ &\quad C(\gamma_{ai}B_{exci})u[k] + Cw[k] + v[k])\\ &= \hat{x}[k+1\backslash k] + g_iC(\gamma_{ai}B_{exci})u[k] + g_i(Cw[k] + v[k])\\ &= \hat{x}[k+1\backslash k] + (\gamma_{ai}B_{exci})u[k] + g_i(Cw[k] + v[k])\end{aligned}$$

$$then\quad x[k+1] \approx \hat{x}[k+1/k+1]$$

$$and\ the\ value\ y[k+1] - C\hat{x}[k+1/k+1] =$$

$$Cx[k] - C\hat{x}[k/k] \approx 0\ or\ at\ least\ less\ than\ the\ threshold$$

But this filter will be sensitive to any change in other actuators e.g. if a fault occurred in the j[th] actuator (j≠i) then

$$x[k+1] = Ax[k] + B_{incj}u[k] + \gamma_{aj}B_{excj}u[k] + w[k]$$

$$y_f[k] = Cx[k] + v[k]$$

$$\hat{x}[k+1\backslash k] = A\,\hat{x}[k\backslash k] + B_{inci}u[k]$$

$$\begin{aligned}\hat{x}[k+1/k+1] &= \hat{x}[k+1\backslash k] + g_i(y_f[k+1] - C\hat{x}[k+1\backslash k])\\ &= \hat{x}[k+1\backslash k] + g_i(C(Ax[k] + B_{incj}u[k] + \gamma_{aj}B_{excj}u[k]\\ &\quad + w[k]) + v[k] - C(A\hat{x}[k/k] + B_{inci}u[k]))\\ &= \hat{x}[k+1\backslash k] + g_i(CA(x[k] - A\hat{x}[k/k) + C(B_{incj}\\ &\quad + \gamma_{aj}B_{excj} - B_{inci})u[k]) + g_i(Cw[k] + v[k])\end{aligned}$$

$$here\ e[k]\ is\ no\ longer\ has\ a\ zero\ mean$$

$$\begin{aligned}&and\ the\ value\ y[k+1] - C\hat{x}[k+1/k+1]\\ &= Cx[k+1] - C\hat{x}[k+1/k+1]\\ &= C(Ax[k] + B_{incj}u[k] + \gamma_{aj}B_{excj}u[k] + w[k]) + v[k]\\ &\quad - C(A\,\hat{x}[k\backslash k] + B_{inci}u[k] + g_i(CA(x[k] - A\hat{x}[k/k)\\ &\quad + C(B_{incj} + \gamma_{aj}B_{excj} - B_{inci})u[k]) + g_i(Cw[k] + v[k]))\\ &= C(I - g_iC)A(x[k] - A\hat{x}[k/k) + C(I - g_iC)B_{incj}u[k]\\ &\quad + \gamma_{aj}C(I - g_iC)B_{excj}u[k] + C(1 - g_i)B_{inci}u[k] + Cg_i(Cw[k] + v[k])\\ &\neq 0\ and\ because\ (I - g_iC) \neq 0\ this\ value\\ &\quad will\ give\ a\ residual\ which\ can\ detect\ a\ \ fault\end{aligned}$$

By this way g_i will be insensitive to any fault in i[th] actuator but it will be sensitive to any other actuator fault, by this way if a filter is used for each fault then when a fault occurred all of these filters will produce a residual except one, a result of this is shown in the simulation results

5.2.3.2- the estimation

Because the uses of residual generated by the state estimators are at high extend affected by the noise and they will give the fault estimation indirectly, this way required a long time to get consistent estimation of the fault [7] then for the fault estimation can be made using parameter estimation techniques. Here the problems that face using parameter estimation for FDD is solved by combining parameters and states estimators because already the state estimator made the fault detection and isolation and we are about to estimate it, then if a filter is used to isolate only one fault then any signal with persistence excitation of degree more than one (which is the case in almost all the input signals) will do the job, then we can use the parameter estimation techniques to identify the fault by treating the other parameter as constants

A very important point here is that in most papers they deal only with a multiplicative fault which is represented by multiplying the system parameters by a factor but there are other types of fault which were discussed earlier then we have to check the following values:

1- The value of b_i (if a fault in the i^{th} actuator is detected and isolated) then if the estimation of this value gives a value that can be represented by $\gamma_{ai}b_i$ this is a multiplicative fault with this effect

2- The value of b_i*u_i if this amount is always fixed then the fault represented by blocked open then in this case whatever the value of u_i the output from the actuator will be always the same

5.2.3.3- the limits of the procedure

-it required a linear (or linearized model) which is not working for systems that don't have a valid linear model

-the system must be able to tolerate some time of less performance before losing its stability or linearity because there is no way to get fault diagnosis in no time some time even if very small is required to get a valuable information

-this procedure is helpful only for a fault in the actuators and the sensors.

- its valid only to detect and isolate one fault at a time.

5.3- control reconfiguration

Here after getting the information about the fault from the FDI and this information contains

1-where did the fault happened

2-the type of the fault (Block open, blocked closed or some deviation)

3- A value of the fault in improving manner.

The third point is very important because we can not wait until the fault estimator get a consistent estimation of the fault, then we want to use the available information, but with a degree of uncertainty e.g. in [7] the author gives a way to get early estimation for the fault value plus limit in which this value can vary within, then by this way the FTC should use this information to preserve the most important properties The proposal here is to make the change in the input signals in case of actuator fault and to use the estimated state to generate the correct output in the sensor fault case then the effect of sensor fault in term of reconfiguration is not a big problem like the actuator one. So our concentration here will be toward faults in actuator

The proposed method is like this

- if the fault is multiplicative fault like $\gamma_{ai}b_i$ and we can multiply the corresponding input by $1/\gamma_{ai}$, and the actuator will not saturate then just to take real input u_i and multiply it by $1/\gamma_{ai}$, by this way the normal situation will be preserved
- if the fault is block fault (Blocked open or Blocked closed) or the fault is multiplicative fault like $\gamma_{ai}b_i$ and the value of $u*(1/\gamma_{ai})$ will saturate the actuator then we have to make a reconfiguration which will be described next.

for illustration a system with three inputs p=3 three states n=3 and the fault occurred in the second actuator will be used:

The old system was

$$\begin{bmatrix} b_{11} & b_{12} & b \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

and the new system will be

$$\begin{bmatrix} b_{11} & \gamma b_{12} & b_{31} \\ b_{21} & \gamma b_{22} & b_{32} \\ b_{31} & \gamma b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix}$$

the values of $\bar{u}_i$ i=1,2,3 can be selected in different ways and this is the general form. Normally the number of states will be greater the number of input then the number of equations will be more than the number of

variables then pseudo-inverse can be used for the solution.
one possible selection is to make

$$\begin{bmatrix} b_{11} & \gamma b_{12} & b_{31} \\ b_{21} & \gamma b_{22} & b_{32} \\ b_{31} & \gamma b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \bar{u}1 \\ \bar{u}2 \\ \bar{u}3 \end{bmatrix}$$

to be as close as possible to

$$\begin{bmatrix} b_{11} & b_{12} & b \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

One approach is every time the values of u_1,u_2 and u_3 are got the values of $\bar{u}_1, \bar{u}_2$ *and* $\bar{u}_3$ will be calculate on line to give a closer result as the nominal one according to the system requirement

If the system can not tolerate delay in the response then the system could be made by combining the inputs that can have similar effects (redundant) like:

$$\begin{bmatrix} b_{11} & \gamma b_{12} & b_{13} \\ b_{21} & \gamma b_{22} & b_{32} \\ b_{31} & \gamma b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_1 + \alpha_{12} u_2 \\ \alpha_{22} u_2 \\ u_3 + \alpha_{32} u_2 \end{bmatrix}$$

Then we have to solve the equations

$$\begin{bmatrix} 1 - b_{12} * \gamma\alpha_{22} \\ 1 - b_{22} * \gamma\alpha_{22} \\ 1 - b_{32} * \gamma\alpha_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{13} \\ b_{21} & b_{23} \\ b_{31} & b_{33} \end{bmatrix} \begin{bmatrix} \alpha_{12} \\ \alpha_{32} \end{bmatrix}$$

Here because there is no fault related to u_1 and u_3 then they will not be affected, u_2 will be multiply by a factor that will give maximum allowed value in gain of the actuator without enter to the saturation mode α_{12} and α_{32} should be selected in away based on optimization analysis to give from the three equation the optimal solution based on some constrains and this analysis is to take place in the design stage that the value of α_{12} and α_{32} are put in equations that relate the fault value to the required modification e.g. to take the least squares then the solution should be

$$\left(\begin{bmatrix} b_{11} & b_{13} \\ b_{21} & b_{23} \\ b_{31} & b_{33} \end{bmatrix}^T \begin{bmatrix} b_{11} & b_{13} \\ b_{21} & b_{23} \\ b_{31} & b_{33} \end{bmatrix} \right)^{-1} \begin{bmatrix} b_{11} & b_{13} \\ b_{21} & b_{23} \\ b_{31} & b_{33} \end{bmatrix}^T \begin{bmatrix} 1 - b_{12} * \gamma\alpha_{22} \\ 1 - b_{22} * \gamma\alpha_{22} \\ 1 - b_{32} * \gamma\alpha_{22} \end{bmatrix} = \begin{bmatrix} \alpha_{12} \\ \alpha_{32} \end{bmatrix}$$

In the case of blocked closed or blocked open then the same procedure will be used but the middle columns of the B matrix will be zero and the fixed amount is to added for each equation

$$\begin{bmatrix} b_{11} & 0 & b_{13} \\ b_{21} & 0 & b_{32} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{bmatrix} u_1 + \alpha_{12} u_2 \\ \alpha_{22} u_2 \\ u_3 + \alpha_{32} u_2 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$$

Theses analysis is based on open loop case for closed loop the modification values of α_{ij} will be the gains of the feedback then these value will affect the eigenvalues of the system then the use of PIM model following or eigenvalue structure with optimization can be used

5.4- the simulation

5.4.1- the assumed faults

Here we made a simulation for the above mentioned system, first the ability of the system with g_1 to isolate a fault in the first actuator is shown in figure 7, the residual for fault in the 1st actuator by using g_1 and the residual for fault in the 2nd actuator by using g_1 and the it is very obvious that first case is not affected at all and the signal is just due to the noise but in the second figure 8 case a very clear bias is got.

To test the FTC system in figure 9 the effect of the fault in the system performance in case of fault in b_2 without any reconfiguration, this fault occurred at t=100 with factor 0.15, in figure 10 is the outputs y_1 y_2 for the case in which we assume that the parameters can be modified without reaching the saturation mode here we notice due to inconsistent estimation in the first period immediately after the fault there a deviation in the output finally in figure 11 in a case that the input cannot be modified without getting to the saturation mode of the actuator then we tried to get the best result here y_2 is got exactly not like the case of y_1 which is deteriorated by an amount which may acceptable or not according to the system requirements

5.4.2- the result

The following figures shows the result of the simulation result, the simulation is made by Matlab using the previous model, here we check the isolation of the fault in the first actuator, and the reconfiguration of a fault in the second actuator

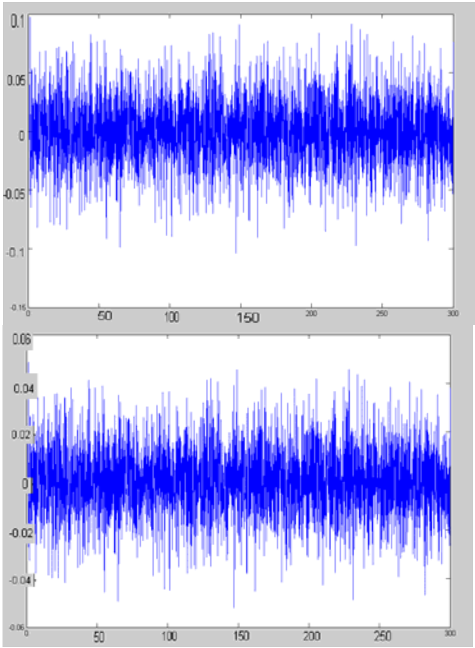

Figure 7 residual g_1 when there is a change by factor 0.15 in b_1 y_1 (top) and y_2 (bottom)

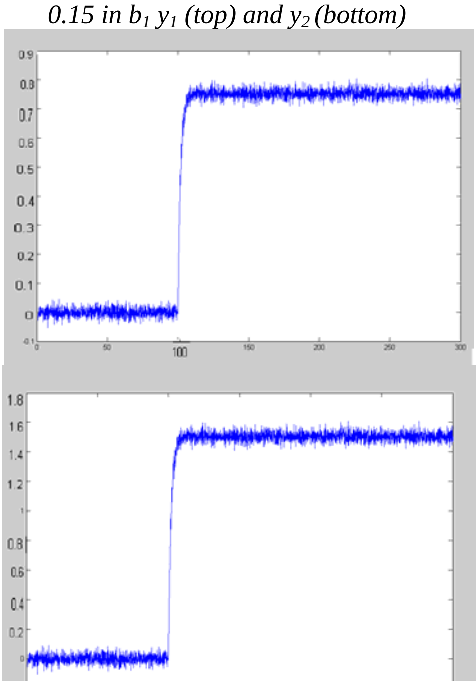

Figure 8 residual g_1 when there is a change by factor 0.15 in b_2 y_1 (top) and y_2 (bottom)

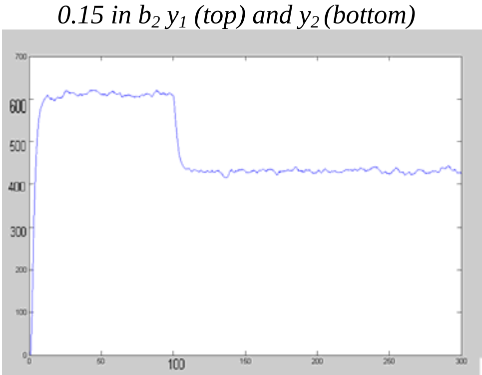

Figure 9(*TOP*)

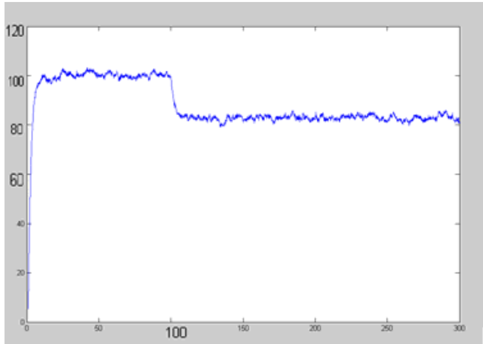

Figure 9 y_1 (top) and y_2 (bottom) after the fault without reconfiguration

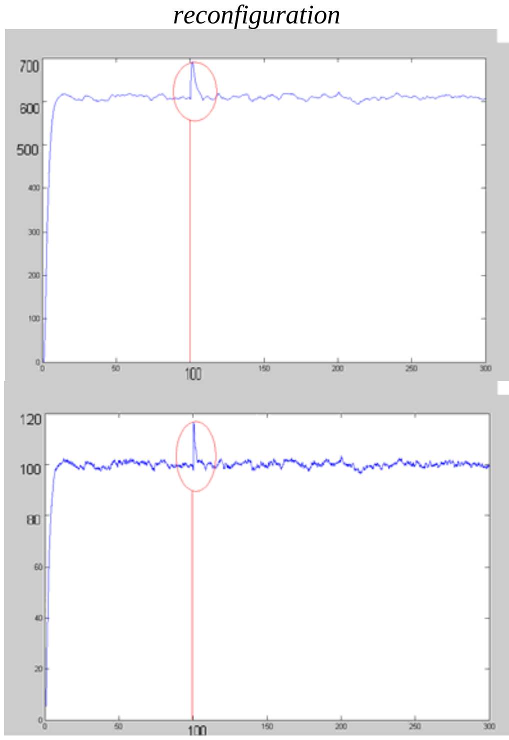

Figure 10 y_1 (top) and y_2 (bottom) with modifying b_2 without saturation

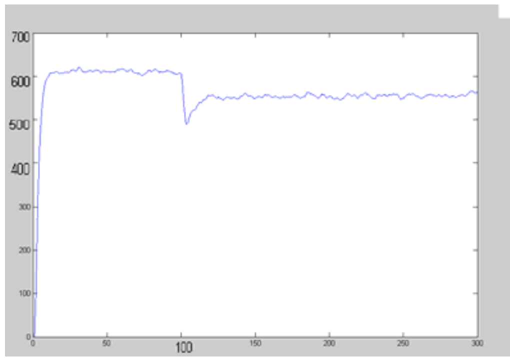

Figure11(*TOP*)

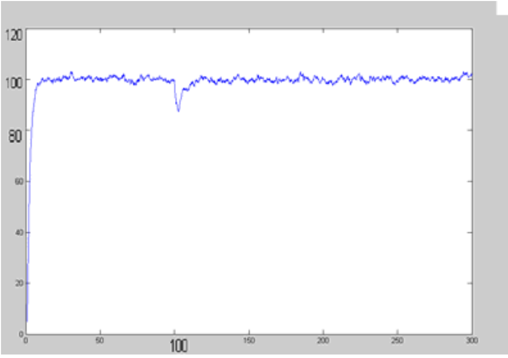

Figure11 y_1 (top) and y_2 (bottom) after modification but with saturated b_2

6-Conclusion

This report is an explanation of the basic concepts, active fields and methods that are required to make fault tolerant control system, to make FTCS its required extensive analysis about the requirement from the FTCS and the available information and redundancy of the system, for how much performance degradation is acceptable in the system and how much delay in response for the fault is tolerated, the faults cause effect analysis with quantitative demonstration, FDD analysis according to the availability of the model in its accuracy and not to forget the rule of the experts in this part, to make control reconfiguration also the basic requirements must be determined and to be put in an order according to their importance, the link between FDD and control reconfiguration in term of accuracy of the result and the time delay must be analyzed to get the acceptable area. Also a modification of using extended constrained Kalman filtering plus parameter estimation techniques are used for FDD, a reconfiguration procedure by updating the way by which the input enter the system is also given.

This field is very active but its application dependent then the better understanding of the system and its requirement the better result we can get. So it is expected that the research in this field will continue actively in the coming years.

References

[1]Y .M.Zhang and J.Jiang" Active fault-tolerant control system against partial actuator failures" IEE Proc.-Control Theory Appl., Vol. 149, No. I , January 2002

[2]Y.M. Chena,, M.L. Lee " Neural networks-based scheme for system failure detection and diagnosis" Elsevier Science Publishers B. V Mathematics and Computers in Simulation 58 (2002) 101–109

[3] Patton, R. J. (1997), "Fault-tolerant control: the 1997 situation", Preprints of IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, Kingston upon Hull, UK

[4] E. Wu, K. Zhou, and G. Salomon, “Control reconfigurability of linear time-invariant systems,” **Automatica**, vol. 36, No. 11, pp. 1767-1771, November 2000.

[5] N. Eva Wu " Coverage in fault-tolerant control" Automatica 40 (2004) 537 – 548

[6] Ying Hu Petros A. Ioannou and Majdedin Mirmirani" Fault-Tolerant Control and Reconfiguration for High Performance Aircraft: Review " CATT Technical Report 01-11-01 University of Southern California Dept. of Electrical Engineering-Systems Los Angeles, CA 90089 California State University, Los Angeles Dept. of Mechanical Engineering Los Angeles, CA

[7] Qing Zhao " FAULT TOLERANT CONTROL SYSTEMS DESIGN" Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Faculty of Graduate Studies The University of Western Ontario London, Ontario 1999

[8] Mogens Blanke Jesper Sandberg Thomsen " Electrical steering of vehicles - fault-tolerant analysis and design" Microelectronics Reliability 46 (2006) 1421–1432

[10] Yonghui Xu and Jin Jiang "Optimal Sensor Location in Closed-Loop Control Systems for Fault Detection and Isolation"Proceedings of the American Control Conference Chicago, Illinois June 2000

[11] Jin Jiang and Youmin Zhang A revisit to block and recursive least squares for parameter estimation *Computers & Electrical Engineering*, *Volume 30, Issue 5, July 2004*, *Pages 403-416*

[12] DANIEL AVITZOUR " Active Fault Tolerant Control with Actuation Reconfiguration" IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS VOL. 40, NO. 3 JULY 2004

[13] Larson, E. C., Parker, B. E., Jr., and Clark, B. R. (2002) Model-based sensor and actuator fault detection and isolation.
In *Proceedings of the American Control Conference*, Anchorage, AK, 2002, 4215—4219.

[14] Blanke, Mogens "Diagnosis and fault-tolerant control" ISBN: 3540010564 Springer 2003

[15] Concepts and Methods in Fault-tolerant Control Blanke, Mogens ; Staroswiecly, M. ; Wu, N.E. in book: Proc. American Control Conference 2001, pages: 2606-2620, 2001, 2001

[16] Bong Chae Moon " FAULT TOLERANT CONTROL BASED ON A **NEW** ACCOMMODATION FILTER" CH2917-3/90/0000-2155$1 **.OO** 1990 IEEE